

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2003-502730

(P2003-502730A)

(43) 公表日 平成15年1月21日 (2003.1.21)

(51) Int.Cl. <sup>1</sup>	級別記号	P I	フールド* (参考)
G 0 6 F	9/54	G 0 6 F	9/08
	9/44		6 4 0 B
	9/445		5 B 0 7 6
			6 1 0 Q
			6 2 0 K

審査請求 未請求 予備審査請求 有 (全 53 頁)

(21) 出願番号 特願2001-503049 (P2001-503049)  
 (86) (22) 出願日 平成12年5月23日 (2000.5.23)  
 (85) 翻訳文提出日 平成13年12月17日 (2001.12.17)  
 (86) 国際出願番号 P C T / U S 0 0 / 1 4 1 7 3  
 (87) 国際公開番号 W O 0 0 / 0 7 7 6 3 3  
 (87) 国際公開日 平成12年12月21日 (2000.12.21)  
 (31) 優先権主張番号 0 9 / 3 3 3 , 4 2 2  
 (32) 優先日 平成11年6月15日 (1999.6.15)  
 (33) 優先権主張国 米国 (US)

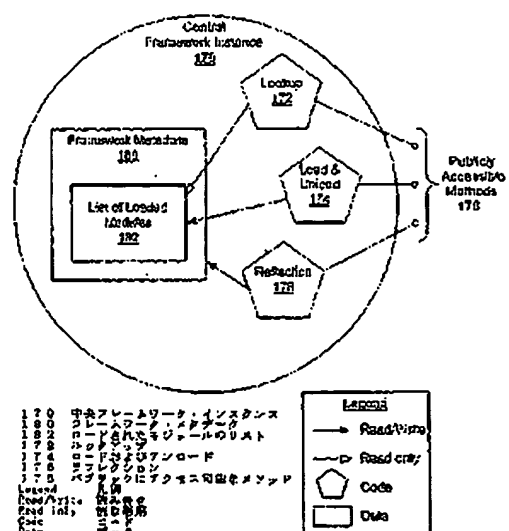
(71) 出願人 サン・マイクロシステムズ・インコーポレイテッド  
 Sun Microsystems, Inc.  
 アメリカ合衆国 カリフォルニア 95054,  
 サンタ クララ, ネットワーク サー  
 クル 4150  
 (72) 発明者 ベイカー, マーク  
 カナダ国・ケイ1エム 2シー5・オンタ  
 リオ・オタワ・マッケイ ストリート・  
 389  
 (74) 代理人 弁理士 山川 政樹

最終頁に続く

(54) 【発明の名称】 ホットプリント・デバイスのためのジャストインタイム・サービス

## (57) 【要約】

共用可能プログラム・モジュールをサポートする軽量包含フレームワークである。上記包含フレームワークは、パーソナル・データ・アシスタント (PDA)、スマート・セルラー電話、全世界測位システム (GPS) レシーバ等、多種多様なリソース制約形ホットプリント・デバイスで動作するには十分コンパクトであり、かつ、有効である。上記包含フレームワークにより、モジュール登録、ルックアップ、インスタンス追跡等のモジュール管理サービスが提供される。フレームワークによって提供される基本機能は、フレームワークにシステム・モジュールを追加することによって拡張することができる。上記共用可能プログラム・モジュールは、サービスを実施することができる。モジュールは、サービス・プロバイダおよび他のサービス・モジュールのクライアントの両方であっても良い。軽量かつ有効なシステムの性質により、ウェブ・ブラウザ、電子メール・クライアント等、従来、デスクトップ・コンピュータ・システムに関連するアプリケーション・プログラムを、ホットプリント・デバイス上で実行することができる。包含フレ



(2)

特表2003-502730

**【特許請求の範囲】**

【請求項1】 フレームワークが共用可能プログラム・モジュールをサポートし、前記プログラム・モジュールが、コンピュータ・サービスの実施を実行することができる、小フットプリント・デバイスのためのソフトウェア・フレームワーク。

【請求項2】 前記小フットプリント・デバイスがオペレーティング・システムを備え、さらに実行可能ソフトウェア環境を備え、それにより前記ソフトウェア環境内のソフトウェア・プログラムの実行を可能にし、また、前記ソフトウェア・フレームワークが、前記ソフトウェア環境内で実行するソフトウェア・コンポーネントを備える請求項1に記載のソフトウェア・フレームワーク。

【請求項3】 前記実行可能ソフトウェア環境がJ a v a（商標）アプリケーション環境である請求項2に記載のソフトウェア・フレームワーク。

【請求項4】 前記ソフトウェア・コンポーネントが、J a v a（商標）クラスのインスタンスである請求項3に記載のソフトウェア・フレームワーク。

【請求項5】 前記ソフトウェア・コンポーネントが、前記プログラム・モジュールを前記ソフトウェア環境中にロードするように動作することができる請求項2に記載のソフトウェア・フレームワーク。

【請求項6】 第1のプログラム・モジュールが前記ソフトウェア環境中にロードされ、前記第1のプログラム・モジュールが、第2のプログラム・モジュールを前記ソフトウェア環境中にロードするよう、前記ソフトウェア・コンポーネントに要求するように動作することができ、前記ソフトウェア・コンポーネントが、前記第1のプログラム・モジュールからの前記要求の受信に応じて、前記第2のプログラム・モジュールを前記ソフトウェア環境中にロードするように動作することができる請求項2に記載のソフトウェア・フレームワーク。

【請求項7】 前記ソフトウェア・コンポーネントがさらに、前記第2のプログラム・モジュールへの参照を前記第1のプログラム・モジュールに戻すように動作することができ、前記第1のプログラム・モジュールが、前記第2のプログラム・モジュールを呼び出すように動作することができる請求項6に記載のソフトウェア・フレームワーク。

(3)

特表2003-502730

【請求項8】 第1のプログラム・モジュールが前記ソフトウェア環境中にロードされ、前記第1のプログラム・モジュールが、前記ソフトウェア・コンポーネントにモジュール記述子を渡すように動作することができ、前記ソフトウェア・コンポーネントが、前記モジュール記述子と一致する第2のプログラム・モジュールを探しだし、かつ、前記第2のプログラム・モジュールを前記ソフトウェア環境中にロードするように動作することができる請求項2に記載のソフトウェア・フレームワーク。

【請求項9】 第1のプログラム・モジュールが前記ソフトウェア環境中にロードされ、前記第1のプログラム・モジュールが、前記ソフトウェア・コンポーネントにモジュール記述子を渡すように動作することができ、前記ソフトウェア・コンポーネントが、前記モジュール記述子と一致する第2のプログラム・モジュールを探しだし、かつ、前記第2のプログラム・モジュールへの参照を前記第1のプログラム・モジュールに戻すように動作することができる請求項2に記載のソフトウェア・フレームワーク。

【請求項10】 第1のプログラム・モジュールおよび第2のプログラム・モジュールが前記ソフトウェア環境中にロードされ、前記第2のプログラム・モジュールが、第3のプログラム・モジュールへの参照を前記ソフトウェア・コンポーネントに要求するように動作することができ、また、前記ソフトウェア・コンポーネントが、前記第3のプログラム・モジュールに対する要求を前記第1のプログラム・モジュールに戻すよう、前記第2のプログラム・モジュールに命令するように動作することができる請求項2に記載のソフトウェア・フレームワーク。

【請求項11】 前記ソフトウェア・フレームワークが、第1のプログラム・モジュールを第2のプログラム・モジュールに対するモジュール要求リスナとして登録するための手段を備え、前記第1のプログラム・モジュールが、前記第2のプログラム・モジュールからの、モジュール記述子と一致する第3のプログラム・モジュールへの参照要求を受け取るように動作することができ、また、前記第1のプログラム・モジュールが、前記モジュール記述子と一致する第3のプログラム・モジュールへの参照を前記第2のプログラム・モジュールに戻すよう

(4)

特表2003-502730

に動作することができる請求項1に記載のソフトウェア・フレームワーク。

【請求項12】 前記ソフトウェア・フレームワークが、前記プログラム・モジュールに関するデータを備え、また、前記プログラム・モジュールが、前記データのすべてにアクセスすることができる第1のクラスのシステム・モジュールを備え、前記プログラム・モジュールがさらに、前記データの少なくとも一部にアクセスすることができない、第2のクラスのモジュールを備える請求項1に記載のソフトウェア・フレームワーク。

【請求項13】 第1のプログラム・モジュールが前記ソフトウェア内で動作し、前記第1のプログラム・モジュールが、第1のコンピュータ・サービスの実施を実行することができ、また、第2のプログラム・モジュールが前記ソフトウェア内で動作し、前記第2のプログラム・モジュールが、第2のコンピュータ・サービスの実施を実行することができ、かつ、前記第2のプログラム・モジュールが、前記第1のプログラム・モジュールを利用して前記第2のコンピュータ・サービスを実施する請求項1に記載のソフトウェア・フレームワーク。

【請求項14】 前記第2のコンピュータ・サービスがウェブ・ブラウジング・サービスであり、前記ウェブ・ブラウジング・サービスが、前記小フットプリント・デバイスのユーザによるウェブ・ページを見ることができるようになる請求項13に記載のソフトウェア・フレームワーク。

【請求項15】 前記小フットプリント・デバイスがシステム・メモリを備え、また、前記ソフトウェア・フレームワークが、1つまたは複数のソフトウェア・コンポーネントを備え、前記1つまたは複数のソフトウェア・コンポーネントが、前記システム・メモリに記憶され、また、前記1つまたは複数のソフトウェア・コンポーネントが、300キロバイト未満のメモリ空間しか必要としない請求項1に記載のソフトウェア・フレームワーク。

【請求項16】 前記小フットプリント・デバイスが、パーソナル・データ・アシスタント(PDA)、セルラー電話、および全世界測位システム(GPS)レシーバからなるグループからの小フットプリント・デバイスである請求項1に記載のソフトウェア・フレームワーク。

【請求項17】 前記小フットプリント・デバイスが、ゲーム・コンソール

(5)

特表2003-502730

、ウェアラブル計算デバイス、セットトップ・ボックス、および電子ブック・デバイスからなるグループからの小フットプリント・デバイスである請求項1に記載のソフトウェア・フレームワーク。

【請求項18】 前記小フットプリント・デバイスが、2メガバイト未満のメモリを備える請求項1に記載のソフトウェア・フレームワーク。

【請求項19】 前記小フットプリント・デバイスが、20平方インチ未満のディスプレイ画面を備える請求項1に記載のソフトウェア・フレームワーク。

【請求項20】 前記小フットプリント・デバイスが、バッテリー駆動専用である請求項1に記載のソフトウェア・フレームワーク。

【請求項21】 プログラム・モジュールを動作させることができる小フットプリント・デバイスであって、前記小フットプリント・デバイスが、  
処理デバイスと、  
システム・メモリと  
を備え、

前記システム・メモリが、オペレーティング・システムおよび実行可能ソフトウェア環境を記憶し、前記ソフトウェア環境中でソフトウェア・プログラムを実行させることができ、

前記システム・メモリがさらに、本発明による1つまたは複数のソフトウェア・コンポーネントを記憶し、前記1つまたは複数のソフトウェア・コンポーネントが、前記ソフトウェア環境内で実行し、かつ、前記1つまたは複数のソフトウェア・コンポーネントが、共用可能プログラム・モジュールをサポートするソフトウェア・フレームワークを実施し、前記プログラム・モジュールが、コンピュータ・サービスの実施を実行することができる小フットプリント・デバイス。

【請求項22】 前記実行可能ソフトウェア環境がJava（商標）アプリケーション環境である請求項21に記載の小フットプリント・デバイス。

【請求項23】 前記ソフトウェア・コンポーネントの1つまたは複数が、前記プログラム・モジュールを前記ソフトウェア環境中にロードするように動作することができる請求項21に記載の小フットプリント・デバイス。

【請求項24】 第1のプログラム・モジュールが前記ソフトウェア環境中

(6)

特表2003-502730

にロードされ、前記第1のプログラム・モジュールが、第2のプログラム・モジュールを前記ソフトウェア環境中にロードするよう、前記ソフトウェア・コンポーネントの1つまたは複数に要求するように動作することができ、前記ソフトウェア・コンポーネントの前記1つまたは複数が、前記第1のプログラム・モジュールからの前記要求の受信に応じて、前記第2のプログラム・モジュールを前記ソフトウェア環境中にロードするように動作することができる請求項21に記載の小フットプリント・デバイス。

【請求項25】 前記ソフトウェア・コンポーネントの前記1つまたは複数がさらに、前記第2のプログラム・モジュールへの参照を前記第1のプログラム・モジュールに戻すように動作することができ、前記第1のプログラム・モジュールが、前記第2のプログラム・モジュールを呼び出すように動作することができる請求項24に記載の小フットプリント・デバイス。

【請求項26】 前記ソフトウェア・フレームワークが、前記プログラム・モジュールに関するデータを備え、また、前記プログラム・モジュールが、前記データのすべてにアクセスすることができる第1のクラスのシステム・モジュールを備え、前記プログラム・モジュールがさらに、前記データの少なくとも一部にアクセスすることができない、第2のクラスのモジュールを備える請求項21に記載の小フットプリント・デバイス。

【請求項27】 前記小フットプリント・デバイスが、パーソナル・データ・アシスタント（PDA）、セルラー電話、および全世界測位システム（GPS）レシーバからなるグループからの小フットプリント・デバイスである請求項21に記載の小フットプリント・デバイス。

【請求項28】 前記小フットプリント・デバイスが、ゲーム・コンソール、ウェアラブル計算デバイス、セットトップ・ボックス、および電子ブック・デバイスからなるグループからの小フットプリント・デバイスである請求項21に記載の小フットプリント・デバイス。

【請求項29】 前記システム・メモリの記憶容量が2メガバイト未満である請求項21に記載の小フットプリント・デバイス。

【請求項30】 前記小フットプリント・デバイスが、バッテリー駆動専用で

(7)

特表2003-502730

ある請求項21に記載の小フットプリント・デバイス。

(8)

特表2003-502730

## 【発明の詳細な説明】

## 【0001】

## (発明の背景)

## (発明の分野)

本発明は、ハンドヘルド・コンピュータ、パーソナル・データ・アシスタント(PDA)、セルラー電話等のリソースに制約のある小フットプリント・デバイスの分野に関する。本発明は、より詳細には、小フットプリント・デバイスにおける共用可能サービスをサポートする軽量フレームワークを含んでいる。

## 【0002】

## (関連技術の説明)

「スマートな」小フットプリント・デバイスの分野は、生長しつつあり、すばやく変化しつつある。小フットプリント・デバイスには、ハンドヘルド・コンピュータ、携帯情報端末(PDA)、セル電話、全世界測位システム(GPS)受信器、ゲーム・コンソール、および多数のそのようなデバイスが含まれる。これらのデバイスは、よりインテリジェントになり、相互接続されつつある。Sun Microsystems, Inc.社のJini(商標)などのテクノロジーおよびOpen Service Gateway Initiative(OSGI)などのイニシアティブによって、コンピュータ・ネットワークの従来の概念が、小フットプリント・デバイスを含むように拡張されつつある。

## 【0003】

このデバイス相互接続の増加によって、デバイス間ベースのサービスとデバイス内ベースのサービスの両方のコンピューティング・サービスを統合する新しいタイプのコンピューティング・サービスと新しい方法の必要性が導かれた。「サービス」とは、個人、アプリケーション、または他のサービスによって使用することのできるデバイス内に実装されるかそのデバイスからアクセス可能であるエンティティである。サービスの概念は、広く、多数の異なるスケールで考慮することができる。たとえば、サービスには、共用印刷、電子メール、電話などの、よく知られたネットワークベースのサービスが含まれる。サービスには、ローカル・ネットワーク内のデバイスの電力消費を制御することができるエネルギー管理



(9)

特表2003-502730

サービス、エラー発生時にデバイスがサービス技術者に情報を送信できるようにする診断サービス、緊急事態について保健専門家に即座に通知する保健監視サービスなどの、より知られていない例も含まれる。

#### 【0004】

サービスには、ローカル・マシンまたはデバイス内に配置され、実行可能であるモジュールまたはアプリケーションも含まれる。たとえば、ローカル・アプリケーション・プログラムが、カレンダー・サービス、連絡先リスト・サービス、ブックマーク・サービスなどを使用することができる。この例では、アプリケーション・プログラムが、これらのサービスを一緒に使用して、ユーザが連絡先リストから人を選択し、その人と会う面会時間を記録し、面会項目に簡単にアクセスするためのブックマークを作成できるようにすることができる。

#### 【0005】

現在、複数のサービスとアプリケーションを単一の小フットプリント・デバイス内で一緒に実行することがますます一般的になりつつある。しかし、一般にメモリ、処理能力、および他のリソースが、小フットプリント・デバイスでは非常に限られるので、サービスとアプリケーションの所望の統合を達成するためには特殊化された軽量ソフトウェア・フレームワークが必要である。また、フレームワークが、すべての種類の小フットプリント・デバイス用のすべての種類のサービスとアプリケーションのサポートを提供するのに十分に柔軟であり拡張可能であることが望ましい。もう1つの目標は、フレームワークが、J i n i (商標) ネットワーク内のデバイスから使用可能なサービスなどのオフデバイス (デバイス外) ・サービスとの互換性を有し、これと統合されることである。小フットプリント・デバイスで動作するアプリケーション/サービスのための軽量で拡張可能なネットワーク・サービス互換の包含フレームワーク (containment framework) を本明細書で説明する。

#### 【0006】

##### (発明の概要)

本発明は、プログラム・モジュールを小フットプリント・デバイス内に含め、かつ、管理するシステムおよび方法を含んでいる。本明細書で使用されているよ

(10)

特表2003-502730

うに、モジュールとは、実行可能なソフトウェア・コンポーネントのことである。本発明は、共用可能モジュールのための包含フレームワークを含んでいる。この包含フレームワークの一実施態様は、York. 1. 1と呼ばれている。包含フレームワークは、モジュール登録、ルックアップ、インスタンス追跡等を可能にする。包含フレームワーク内のモジュールを、サービスとして他のモジュールが使用することができる。包含フレームワークは動的であり、必要に応じてモジュールを登録し、ロードすることができる。

## 【0007】

モジュールは、パーセルにグループ分けし、ユニットとして管理することができる。パーセルは、いくつかの目的を果たすことができる。例えば、モジュールの実行時コンテキストのほとんどをパーセルが決定し、そこから発信することができる。また、インストレーション手続きおよび更新手続きを始め、アプリケーションをパーセルが管理することができる。さらに、パーセル・システムは、多目的開発のサポートを可能にしている（すなわち、開発グループは、パーセルを用いることにより、共通ライブラリに対する信頼を損なうことなく、異なるプラットフォームすなわちシステムを同時に開発することができる）。

## 【0008】

上で説明したように、小フットプリント・デバイスの包含フレームワークは軽量でなければならない。本発明の包含フレームワークは、非常にわずかなメモリを有するデバイスで機能することができる。たとえば、一実施態様では、包含フレームワークが、300KBのみの言込可能メモリを備えたデバイスで機能し、かつ、複数のモジュールが動作するのに十分なメモリ・スペースを残すことができる。さらに、包含フレームワークは、16MHzクラスのチップを有する小フットプリント・デバイスなどの低処理能力を有するデバイスに応答することができる。

## 【0009】

その軽量状態の実現を促進する包含フレームワークの一態様は、包含フレームワークが、非階層型フレームワークであることである。階層包含(hierarchical containment)フレームワークは、フレームワークにおける極めて有効な機能であ

(11)

特表2003-502730

り、モジュールをグループ化するための標準方法を提供し、グループによる他の各サービスの検出およびその使用を可能にしている。しかし、階層包含は、通常、階層包含を実施するために必要なリソースの点に関して、その実現には費用がかかっている。本発明による包含フレームワークにより、通常のオーバーヘッドを必要とすることなく、階層型フレームワークに関連する多くの利点が提供される。モジュールは、他のモジュールによるモジュール・ルックアップ要求を中断することができ、それにより、モジュールが、要求元モジュールの環境をカスタマイズできるようにしている。

## 【0010】

包含フレームワークは、一般的な標準規格に基づくものとすることができる。たとえば、一実施態様では、包含フレームワークは、Pure Java（商標）で記述され、Personal Java（商標）3.0アプリケーション環境に完全に準拠し、その環境で実行できる。Personal Java（商標）は、家庭、オフィス、およびモバイル・ユーザ用の消費者デバイスのために特に設計された、Java（商標）アプリケーション環境である。これには、Java（商標）バーチャル・マシン（JVM）と、コアAPIおよびオブジェクトAPIおよびクラス・ライブラリを含むJava（商標）アプリケーション・プログラミング・インターフェース（API）のサブセットが含まれる。さらに、Personal Java（商標）APIには、リソースを制限された環境で消費者アプリケーションが必要とする特定の機能が含まれる。包含フレームワークを、ハードウェアROMに含めることもでき、ネイティブ・コードにコンパイルすることもできることに留意されたい。

## 【0011】

包含フレームワークは、一般的な標準規格に基づくものとすることができるので、異なるデバイス・タイプおよび異なるベンダによって製造されるデバイスに簡単に移植でき、これによって、タイムトゥーマーケット（time-to-market；市場に出るまでの時間）および開発コストが大幅に削減される。このフレームワークの拡張可能なアーキテクチャによって、異なるデバイスまたはサービスの必要または望みに応じて新しいモジュールをフレームワークに導入することも可能に

(12)

特表2003-502730

なる。このアーキテクチャでは、カスタマイズ可能でスケーラブルなユーザ・インターフェースも可能になる。たとえば、アプリケーションのユーザ・インターフェース・コンポーネントを、異なるデバイスの表示タイプに適當になるようにスワップ・アウトすることができる。

#### 【0012】

下でさらに詳細に説明するように、この包含フレームワークでは、ロードされたモジュールのリストならびに関連するシステム・データが維持される。モジュールは、通常は、たとえば中央フレームワーク・インスタンスのメソッドを呼び出すことによって、間接的にのみモジュール・リストまたはシステム・データにアクセスする。しかし、システム・モジュールと称するいくつかのモジュールは、システム・データ自体にアクセスするか修正する能力を有する。コア・データへのこの種の直接データ・アクセスは、包含フレームワークを軽量に保つのに役立つ。システム・モジュールは、コア・システム・データにアクセスできるので、中央フレームワーク・インスタンスによって実行される基本的なモジュール管理を超える必要なまたは所望の拡張された機能性を、適切なシステム・モジュールを追加することによってシステムに導入することができる。したがって、この包含フレームワークは、非常に拡張可能でもある。

#### 【0013】

システムに、他のモジュールが使用するために使用可能なコア・サービス・モジュールの組を含めることができる。これらのコア・サービスには、上の例で説明した、カレンダー・サービス、連絡先リスト・サービス、およびブックマーク・サービスなどのサービスを含めることができる。そのようなコア・サービスと共に、包含フレームワークは、アプリケーションとサービスの統合されたスートを小フットプリント・デバイス上で実行するための完全なアーキテクチャを提供する。たとえば、Sun Microsystems, Inc. 社から入手可能な Personal Applications スートは、包含フレームワークの一実施態様を中心として構築される。Personal Applications スートには、Personal Applications Browser、Personal Applications Email Client

(13)

特表2003-502730

、およびPersonal Organizerを含む、コンパクトでメモリ効率のよいアプリケーションの統合された組が含まれる。

## 【0014】

本発明のシステムおよび方法を、小フットプリント・デバイスで動作するモジュールの管理に関連付けることができるが、本発明のシステムおよび方法が、この用途に制限されないことに留意されたい。逆に、本発明の多くの態様は、そのように大きいリソース制約を有しないシステムで有用になる可能性がある。その一方で、包含フレームワークの実施態様は、上で説明した小フットプリント・デバイスより大きいリソースの制約を有する環境に簡単に移行可能になるように設計することができる。たとえば、包含フレームワークの実施態様、わずかな修正または無修正で、Embedded Java（商標）アプリケーション環境で動作することができる。

## 【0015】

本発明のその他の目的および利点は、以下の詳細説明を読むことにより、また、添付の図面を参照することにより、明らかになるであろう。

## 【0016】

本発明は、様々な改変および代替態様が可能であるが、実施形態として特定の実施態様を図面に示し、かつ、本明細書において詳細に説明する。しかし、図面および詳細説明は、開示する特定の態様に本発明を制限することを意図したものではなく、すべての改変、等価物および代替態様を、特許請求の範囲の各クレームによって定義されている本発明の精神および範囲の範疇に包含することを意図していることを理解しなければならない。

## 【0017】

(発明の詳細な説明)

## 図1 - ハードウェア・アーキテクチャ・ブロック図

図1は、通常の小フットプリント・デバイスのハードウェア・アーキテクチャを示すブロック図である。本明細書で使用される小フットプリント・デバイスとは、プロセッサやシステム・メモリなどのコンピューティング・リソースを含むが、通常のデスクトップ・コンピュータが有する制約よりかなり大きい、これら

(14)

特表2003-502730

のリソースに対する1つまたは複数の制約を有するハードウェア・デバイスである。たとえば、小フットプリント・デバイスが、2メガバイト以下のメモリを有する場合があるが、通常のデスクトップ・システムは、64メガバイト以上を有する。また、通常の小フットプリント・デバイスは、プロセッサのタイプに関して、プロセッサの速度に関して、またはその両方に関して、通常のデスクトップ・コンピューティング・システムよりかなり低い処理能力を有するであろう。たとえば、携帯情報端末デバイスは、16MHzプロセッサを有するが、通常のデスクトップ・システムは、100MHz以上のプロセッサ速度を有する。また、通常の小フットプリント・デバイスは、デスクトップ・コンピューティング・システムの表示画面よりかなり小さい表示サイズを有する。たとえば、ハンドヘルド・コンピュータの表示画面は、デスクトップ・モニタの表示画面と比較して、通常は小さい。

## 【0018】

示された特定の数は、例示のみであり、比較の目的に使用されることに留意されたい。たとえば、8メガバイト以上のメモリを有する携帯情報端末は、上で示した2メガバイトという通常の数字より多くのメモリを有するが、それでも小フットプリント・デバイスである。

## 【0019】

小フットプリント・デバイスは、上で説明したメモリ、プロセッサ、および表示サイズのリソースのほかに、通常のデスクトップ・コンピューティング・システムと比較して、他のリソース・タイプに対する制約もあるかもしれない。たとえば、通常の小フットプリント・デバイスは、ハード・ディスクを持たない場合があり、ネットワークの接続ができない場合があり、断続的にネットワークに接続できる場合があり、無線ネットワーク接続を有する場合もあるなどである。

## 【0020】

多くの小フットプリント・デバイスは、ポータブルであるか、デスクトップ・コンピュータと比較して小さいが、必ずそうである必要はない。また、多くの小フットプリント・デバイスは、主にまたは排他的に電池動作である。また、小フットプリント・デバイスは、通常は、通常のデスクトップ・コンピューティング

(15)

特表2003-502730

・システムより制限された、または狭い範囲の使用可能性を有する。小フットプリント・デバイスには、ハンドヘルド・コンピュータ、ウェアラブル・デバイス（たとえば腕時計コンピュータ）、携帯情報端末（PDA）、「スマート」セル電話、セットトップ・ボックス、ゲーム・コンソール、全世界測位システム（GPS）ユニット、電子書籍デバイスなどが含まれるが、これに制限はされない。新しい種類の消費者デバイスがすばやく現れているので、小フットプリント・デバイスの網羅的なリストを提供することは不可能である。しかし、用語「小フットプリント・デバイス」は、上で説明した用語の趣旨および範囲内に正当に含めることができるデバイスを含むことが意図される。

#### 【0021】

図1に、通常の小フットプリント・デバイスのブロック図を示す。小フットプリント・デバイスは、望みに応じてさまざまな異なるアーキテクチャを有するものとして行うことができることに留意されたい。本発明のオペレーションを理解するのに必要でないハードウェア要素は、図を単純にするために省略した。

#### 【0022】

図1からわかるように、小フットプリント・デバイスには、プロセッサ100が含まれる。プロセッサ100は、たとえばPentium（登録商標）クラスなどのx86プロセッサ、PowerPCプロセッサ、ならびに他のより能力の低いプロセッサまたは特に小フットプリント・デバイス用に開発されたプロセッサを含む、さまざまなタイプのどれにすることもできる。プロセッサ100は、デスクトップ・コンピュータクラスのプロセッサに見られるものに類似するクロック速度ならびに16MHzなどのより低い速度を含む、さまざまなクロック速度を有することができる。

#### 【0023】

図1からわかるように、デバイスには、システム・メモリ102が含まれる。システム・メモリ102には、RAMまたはROMを含むさまざまなタイプのメモリを含めることができる。通常の小フットプリント・デバイスは、通常のデスクトップ・コンピュータ・システムと比較して、メモリ・ストレージ容量は非常に小さいであろう。

(15)

特表2003-502730

## 【0024】

小フットプリント・デバイスに、1つまたは複数の入力機構を含めることができる。入力機構104が、図1に示されている。入力機構104は、特定のデバイスに適切なさまざまなタイプのいずれかとすることができる。たとえば、入力機構は、キーパッド、マウス、トラックボール、タッチ・ペン、マイクロホンなどとすることができる。

## 【0025】

小フットプリント・デバイスに、1つまたは複数の表示機構を含めることもできる。ディスプレイ106が図1に示されている。しかし、小フットプリント・デバイスに、ディスプレイを含めないか、オーディオ・スピーカなどの別のタイプの出力機構を含めることができる。表示機構106は、特定のデバイスに適切なさまざまなタイプのいずれかとすることができる。スマート・セル電話などの通常の小フットプリント・デバイスの表示機構は、デスクトップ・コンピュータ・システムのディスプレイと比較して小さいものであろう。

## 【0026】

## 図2 - ハードウェア／ソフトウェア階層の図

図2に、包含フレームワーク内でアプリケーションとサービスを実行するシステムで用いられるハードウェア／ソフトウェア層の通常の間層を示す。この図は、例示的であり、さまざまな間層を、特定のデバイスまたは実装に適するように追加するか、組み合わせるか、省略することができる。

## 【0027】

図2に示されたベース層は、デバイス・ハードウェア層120であり、これには、プロセッサおよびシステム・メモリなど、ソフトウェア・システムをサポートするのに必要なハードウェア・リソースが含まれる。一実施形態では、図1に示された小フットプリント・デバイス・ハードウェアの例などの小フットプリント・デバイスのハードウェアによって、図2に示されたハードウェア層120が実装される。しかし、他の実施形態では、ハードウェア層120を、スマート・カードなど、通常の小フットプリント・デバイスよりさらに厳しいリソース制約を有するデバイスなどの、他のタイプのデバイスで実装することができる。



(17)

特表2003-502730

## 【0028】

図2からわかるように、ハードウェア層の上の次の層は、オペレーティング・システム層122である。当技術分野で周知のように、オペレーティング・システムは、デバイス・ハードウェアとそのデバイスで動作するソフトウェアの間のインターフェース層として機能し、入出力、メモリ管理などの低水準タスクのマネージャとして働く。図2に示されたオペレーティング・システム122は、図2に示された上位層をサポートする特定のオペレーティング・システムとすることができる。オペレーティング・システム122は、小フットプリント・デバイスでの使用に適するか、特にそのために記述された、小さく効率的なオペレーティング・システムとすることができる。たとえば、オペレーティング・システム122は、Sun Microsystems, Inc. 社から入手可能なJava（登録商標）OSオペレーティング・システムとすることができる。

## 【0029】

一実施形態では、包含フレームワークが、1つまたは複数のJava（商標）クラスとして、Java（商標）アプリケーション環境に実装される。図2からわかるように、Java（商標）バーチャル・マシン層124およびJava（商標）アプリケーション・プログラミング・インターフェース（API）クラス・ライブラリ層126が、オペレーティング・システムの上の次の層である。この2つの層が、一緒に、Java（商標）アプリケーション環境またはJava（商標）プラットフォームを構成する。包含フレームワークを実施するクラスを、Java（商標）ライブラリ126を使用して構築し、バイトコードにコンパイルすることができる。バイトコードとは、Java（商標）バーチャル・マシン124上で実行され、オペレーティング・システム122またはデバイス・ハードウェア120もしくはその両方と相互作用する命令である。

## 【0030】

一実施形態では、包含フレームワークが、Personal Java（商標）Java（商標）アプリケーション環境で実施され、このPersonal Java（商標）Java（商標）アプリケーション環境は、非常にスケーラブルであり、モジュラーであり、構成可能でありながら、最小限のシステム・リソース

(18)

特表2003-502730

を必要とするように設計されたJava（商標）プラットフォームである。Personal Java（商標）には、Java（商標）パーチャル・マシンと、コアAPI、オプションのAPI、およびクラス・ライブラリを含むJava（商標）APIのサブセットが含まれる。さらに、Personal Java（商標）APIには、Java（商標）抽象ウィンドウ・ツールキット（AWT）の特殊化されたバージョンなどの、リソースを制限された環境で消費者アプリケーションが必要とする特定の機能が含まれる。Personal Java（商標）AWTライブラリは、消費者製品のルック・アンド・フィールを対象とし、それに合わせてチューニングされ、グラフィックス機能およびウィンドウイング機能を提供すると同時に、低解像度ディスプレイおよび代替入力デバイスをサポートする（マウスなしまたはキーボードなしのデバイスのための拡張されたイベント・モデルを介する）。

#### 【0031】

もう一度図2を参照すると、包含フレームワーク128が、Java（商標）プラットフォーム層の上の次の層として図示されている。上で注記したように、包含フレームワーク128は、他のプラットフォームに基づくものとすることもできる。下で詳細に説明するように、包含フレームワーク128は、たとえばモジュール登録、ルックアップ、インスタンス追跡などを可能にすることによって、プログラム・モジュールを管理する。モジュールは、さまざまなサービスを提供することができる。包含フレームワーク128は、モジュールが他のモジュールのサービスを使用するために、モジュールが他のモジュールに要求できるようにする。アプリケーションは、他のモジュールのサービスを使用するモジュールとして実施することができる。したがって、包含フレームワーク128は、軽量で拡張可能なサービスとアプリケーションのフレームワークを提供し、アプリケーションが、共存し、モジュラー・コード・ベースを共用できるようにする。

#### 【0032】

複数のプログラム・モジュールが協力できるようにするこのタイプの拡張可能なアーキテクチャは、小フットプリント・デバイスにとって重要な開発である。小フットプリント・デバイスは、歴史的に、比較的狭い用途に制限されてきた。

(19)

特表2003-502730

たとえば、セル電話は、通常は、電話に使用され、それ以外にはほとんど使用されない。しかし、小フットプリント・デバイスが「よりスマートに」なることを可能にし、汎用プロセッサ、より大きい表示画面などを有するようにするさまざまなテクノロジーが開発されるにつれて、小フットプリント・デバイスで使用されるアプリケーションの範囲を拡張することが望ましくなってきた。

### 【0033】

本発明の包含フレームワークは、一般にデスクトップ・コンピューティング環境に関連するタイプのアプリケーションとサービスを、デスクトップ・コンピュータ・ユーザが慣れている形で、小フットプリント・デバイスと一緒に働かせることができる。図2に示され、上で説明したように、小フットプリント・デバイスで動作するサービスとアプリケーション130を、包含フレームワーク層128上で構築されるモジュールとして実施することができる。たとえば、Sun Microsystems, Inc. 社から入手可能なPersonal Applications スーツは、包含フレームワーク128の一実施形態を使用して構築される。Personal Applications Suiteには、ブラウザ、電子メール・クライアント、およびパーソナル・オーガナイザなどのアプリケーションの統合された組が含まれる。

### 【0034】

図2には、包含フレームワーク128のいくつかの実施形態の、オフデバイス・サービス132をオンデバイス・アプリケーション／サービス130と統合する能力も示されている。たとえば、包含フレームワーク128は、小フットプリント・デバイスと、Jini（商標）ネットワークなどのネットワークの間のインターフェースを提供することができる。小フットプリント・デバイス・システムは、ネットワーク内の他のデバイスまたはクライアントによる使用のために、そのサービスを登録することができる。包含フレームワークは、小フットプリント・デバイス内のサービスとアプリケーションが、他のネットワーク・デバイスによって提供されるサービスをルック・アップでき、使用できるようにすることもできる。小フットプリント・デバイスのサービスの、ネットワーク・サービスとの統合を図3に関して下で詳細に説明する。

(20)

特表2003-502730

## 【0035】

図3から7：例示的なネットワーク・デバイスおよびサービス・フェデレーション(Service Federation)

図3に、包含フレームワーク内でアプリケーション／サービスを動作させる小フットプリント・デバイスが、ローカル・サービススペース・ネットワークに接続される例示的なネットワークを示す。図示の例では、包含フレームワーク144を使用するスマート・セル電話134がネットワークに接続されている。やはりネットワークに接続されて図示されているのが、プリンタ130およびインターネット対応テレビジョン132である。この例では、プリンタ130およびテレビジョン132デバイスが、ネットワークにサービスをエクスポートするように動作可能であり、おそらくはネットワーク上の他のデバイスのサービスを使用すると仮定する。たとえば、プリンタが、その印刷サービス138をエクスポートすることができ、インターネット・テレビジョンが、印刷サービスをルック・アップし、それを使用してウェブ・ページを印刷することができる。この形でのデバイスおよびサービスのフェデレーションを容易にするために、ルックアップ・サービス136が、ネットワーク上に配置される。ルックアップ・サービス136は、ネットワーク・サーバなどの別のデバイスに常駐することができる。

## 【0036】

デバイスおよびサービスのフェデレーションは、さまざまな形で実施することができる。たとえば、Sun Microsystems, Inc. 社から入手可能なJini（商標）テクノロジーに、図3に示されたタイプの分散システムを可能にするコンポーネントおよびプログラミング・モデルが含まれる。一実施形態では、図3に示されたローカル・ネットワークを、Jini（商標）ネットワークとすることができ、プリンタ130およびインターネット・テレビジョン132を、Jini（商標）対応デバイスとすることができる。各デバイスは、Jini（商標）ネットワーク・ルックアップ・サービスを見つけるように動作可能であり、各デバイスが提供するサービスをそのルックアップ・サービスに登録する。ルックアップ・サービスは、サービスによって提供される機能を示すインターフェースを、そのサービスを実装したオブジェクトの組にマッピングする。

(21)

特表2003-502730

## 【0037】

サービスをサービス・フェデレーションに追加するために、デバイスまたは他のサービス・プロバイダは、まず、「ディスカバリ」プロトコルを使用することによって、適切なルックアップ・サービスを突き止めることができる。図4に、ディスカバリ・プロセスを示す。図からわかるように、サービス・プロバイダ164、たとえば図3に示されたプリンタ130は、ローカル・ネットワーク上であらゆるルックアップ・サービスへそれらを識別することを求める要求をブロードキャストすることができる。

## 【0038】

サービス・プロバイダ164が、ルックアップ・サービス160を突き止めた後に、サービス・プロバイダ164は、「ジョイン」プロトコルを使用することによって、そのサービスをルックアップ・サービス160に登録することができる。図5に、ジョイン・プロセスを示す。サービス・プロバイダ164は、クライアントがサービスを呼び出すのに使用することができるサービス・オブジェクトを作成することができる。図5からわかるように、提供されるサービスのサービス・オブジェクトを、提供されるサービスのタイプまたは名前に関する情報を含むサービス属性またはサービス記述子と共に、ルックアップ・サービス160にロードすることができる。たとえば、J i n i（商標）ネットワークでは、図3に示されたプリンタ130が、印刷サービス138のJ a v a（商標）プログラミング・インターフェースを含むサービス・オブジェクトを作成することができる。プリンタ130は、その後、ルックアップ・サービス136の「登録」メソッドを呼び出し、このサービス・オブジェクトを、登録されるサービス138が印刷サービスであること、印刷解像度、可能な用紙サイズなどを指定する属性と共に渡すことができる。

## 【0039】

サービス・プロバイダ164が、そのサービスをルックアップ・サービス160にジョインした後に、他のネットワーク・クライアントが、そのサービスを要求し、使用することができる。サービスを要求するプロセス、ルックアップと称されている、を図6に示す。ルックアップ・サービスを発見した後に、クライア

(22)

特表2003-502730

ント162は、要求されるサービスの記述を使用して、ルックアップ・サービス160にサービスを要求することができる。ルックアップ・サービス160は、リクエストによって与えられた記述を、ルックアップ・サービスにジョインしたサービスと突き合わせることを試みる。ルックアップ・サービス160は、この突合せを実行するために、ジョイン・プロセス中にサービス・プロバイダ164によって送られたサービス属性を使用することができる。一致が見つかった場合に、ルックアップ・サービス160は、適切なサービス・オブジェクトをクライアント162に供給する。たとえば、要求されたサービスのJava（商標）インターフェースをクライアント162に供給することができる。

#### 【0040】

クライアント162は、ルックアップ・サービスからサービス・オブジェクトを受け取った後に、そのサービスを呼び出すことができる。図7に、サービス呼出しのプロセスを示す。サービスが呼び出される時に、クライアント162とサービス・プロバイダ164は互いに直接に通信することができる。この通信には、さまざまな対話プロトコルのどれでも使用することができる。たとえば、使用されるプロトコルを、Java（商標）リモート・メソッド呼出し（RMI）、CORBA、DCOMなどとすることができる。クライアントがルックアップ・サービスから受け取るサービス・オブジェクトが、たとえばRMIメソッドを呼び出すことによって、サービス・プロバイダに配置されたコードをコール・バックすることができ、ローカルに実行して要求されたサービスを提供することができる。これらの手法の組み合わせを使用することができる。

#### 【0041】

図3からわかるように、ローカル・ネットワークのルックアップ・サービス136は、インターネット154などの外部ネットワークへのゲートウェイとして働くこともできる。したがって、このサービスベースの分散コンピューティング・モデルを、ローカル・ネットワークの外部に配置されたクライアントやサービスを含むように拡張することができる。たとえば、Open Service Gateway Initiative（OSGI）用開発されたテクノロジーを活用して、このタイプの分散コンピューティング・システムを実施することが

(23)

特表2003-502730

できる。

#### 【0042】

異なるネットワークやインターネットの間およびそれらにまたがるこのタイプのサービス共用によって、新しいタイプのアプリケーションを開発できるようにすることができる。たとえば、商人が、インターネット・サービスを使用して、特定の顧客に関するデータを記録し、広告サービス・プロバイダがそのデータを使用して、どのローカル・ネットワークにデバイスが接続されているかに応じて、コンテキスト固有の広告を消費者デバイスにプッシュするなどが可能である。たとえば、顧客が、ショッピング・モールに入り、無線接続を介してそのショッピング・モールのローカル・ネットワークに携帯情報端末(PDA)を接続することができる。インターネットベースの消費者データ・サービスが、そのショッピング・モール・ネットワークのルックアップ・サービスにジョインすることができ、モール・ネットワークに接続したばかりの特定の消費者に関する情報を提供することができる。ショッピング・モール・ネットワークで動作するサービスが、パーソナライズされた広告を生成し、顧客のPDAにプッシュするために、顧客のモール内での現在位置、時刻などの他の要因と共にこのデータを使用することができる。

#### 【0043】

図3のネットワークに基づく多数の他のサービスの例が可能である。たとえば、家庭内のネットワーク対応消費者デバイスが、インターネットを介して、電力会社によって提供される、家庭内の電力消費を管理するサービスを使用することができ、セキュリティ・サービス・プロバイダが、ネットワーク・サービスを介して家庭のまたは特定のデバイスを監視することができ、所有地に侵入された時に即座に所有者に通知することができ、保健サービス・プロバイダが、医療機器と通信することによって、患者の状態を遠隔監視することなどができる。

#### 【0044】

上に一覧を示した例では、デバイスが、ネットワークに透過的に接続でき、ネットワーク・サービスをデバイス常駐サービスと統合でき、ネットワーク・クライアントによる使用のためにデバイス常駐サービスをエクスポートできるという

(24)

特表2003-502730

前提が設けられている。本明細書に記載の包含フレームワークは、携帯情報端末、ハンドヘルド・コンピュータ、スマート・セル電話などの小フットプリント・デバイスのサービスとアプリケーションをネットワーク・サービス・フェデレーションと統合するのに必要なインターフェースを提供することができる。

#### 【0045】

図3に示され、下で詳細に説明されるように、包含フレームワーク144は、それ自体のタイプのルックアップ・サービス146を有する。包含フレームワーク144内のルックアップ・サービス146は、上で説明したローカル・ネットワーク・ルックアップ・サービスに類似する形で、ディスカバリ、ジョイン、ルックアップ、およびサービス呼出しのプロセスを使用して動作することができる。たとえば、パーソナル・オーガナイザ・アプリケーション152が、カレンダー・サービス、連絡先リスト・サービス、ブックマーク・サービスなど（図示せず）のさまざまなサービスを使用することができる。パーソナル・オーガナイザ・アプリケーション152は、包含フレームワークのルックアップ・サービス146を介して、これらのサービスと通信するための参照を得ることができる。

#### 【0046】

包含フレームワーク144は、それ自体のルックアップ・サービス146を、図3に示されたローカル・ネットワークのルックアップ・サービス136などのオフデバイスのルックアップ・サービスと統合することもできる。この形で、印刷サービス138およびウェブ・サービス140などのオフデバイス・サービスを、包含フレームワークのアプリケーション／サービス148、150、および152から使用可能にすることができ、その逆も可能である。たとえば、パーソナル・オーガナイザ・アプリケーション152が、包含フレームワークのルックアップ・サービス146に印刷サービスを要求することができる。包含フレームワークのルックアップ・サービス146は、まず、オンデバイスの印刷サービスを探すことができる。見つからない場合には、包含フレームワークのルックアップ・サービス146は、ネットワークのルックアップ・サービス136に印刷サービスを要求することができる。その後、印刷サービス138のサービス・オブジェクトをパーソナル・オーガナイザ152に返すことができる。オンデバイス



(25)

特表2003-502730

のサービス／アプリケーションとオフデバイスのサービスの間のインターフェース142が図3に示されている。オンデバイス／オフデバイスのサービスの統合を実施できる方法に関する詳細をこの後に続ける。

#### 【0047】

上で注記したように、サービスのクライアントは、それ自体が他のクライアントへのサービスになることができる。たとえば、図3に示されたスマート・セル電話の電子メール・クライアント「アプリケーション」150は、それ自体が、包含フレームワーク144内で動作するクライアントまたはネットワーク・クライアントに対するサービスになることができる。たとえば、誤動作の場合に、図3に示されたプリンタ130が、電子メール・サービスを要求し、その結果、診断情報をサービス技術者に送信できるようになる。ネットワークのルックアップ・サービス136は、ネットワークベースの電子メール・サービスを見つけることができない場合に、インターフェース142を介してスマート・セル電話134に電子メール・サービスを要求することができる。包含フレームワーク144内で動作する電子メール・アプリケーション／サービス150のサービス・オブジェクトを要求元のプリンタ・クライアント130に渡すことができる。この例では、プリンタ・クライアント130は、プリンタ・サービス技術者への診断情報を含む電子メールを送信するために、電子メール・アプリケーション／サービス150と直接に通信することができる。電子メール・アプリケーション／サービス150は、電子メール・サーバ・サービスを見つけることができる場合に即座に電子メールを送信することができ、また、後に、セル電話のユーザが異なるネットワークに接続する時にそのようなサービスが使用可能になる時に、電子メールを送信することができる。

#### 【0048】

上の説明では、J i n i（商標）テクノロジーなどの特定のプロトコルおよびプログラミング・モデルを参照したが、これらの特定のテクノロジーが、例示にすぎないことに留意されたい。たとえば、包含フレームワーク内のアプリケーションとサービスを、J i n i（商標）、CORBA、COM／DCOM、Bluetooth、CAL、CEBus、HAVi、Home API、HomePNA

(26)

特表2003-502730

、Home PnP、Home RF、VESAなどを含むがこれに制限されない、さまざまなタイプの標準規格、プロトコル、およびプログラミング・モデルのいずれかを使用する、クライアント、サービス、デバイス、ネットワークなどと統合することができる。

#### 【0049】

##### 図8 ー 包含フレームワークのブロック図

図8は、包含フレームワーク環境の基本アーキテクチャを示す抽象ブロック図である。上で説明したように、包含フレームワークは、アプリケーションとサービスの包含システムを提供する。これらのアプリケーションとサービスは、モジュールと称するユニットとしてシステム内で管理される。包含フレームワークは、軽量であり、一実施形態では、モジュールが、すべてのモジュール管理を実行する単一のフレームワーク・マネージャ・オブジェクトと相互作用することができる。このマネージャを、本明細書では中央フレームワーク・インスタンスと称する。一実施形態では、中央フレームワーク・インスタンスを、Java（商標）クラスのインスタンスとして実施することができる。図8に、中央フレームワーク・インスタンス170と、それに含まれるかそれによって管理されるコードおよびデータを示す。図8に、包含フレームワークの一実施形態が示されていることに留意されたい。他の実施形態は、異なるアーキテクチャを使用することができ、かつ／または、異なるプログラミング言語またはソフトウェア環境で実施することができる。たとえば、図8に示された中央フレームワーク・インスタンス170によって実行されるモジュール管理／包含は、他の実施形態では複数のオブジェクトまたはコンポーネントによって実行することができる。

#### 【0050】

図8からわかるように、中央フレームワーク・インスタンス170には、システムに現在ロードされているモジュールを表すデータ182が含まれる。包含フレームワーク・アーキテクチャは非階層的である。したがって、ロードされたモジュールは、モジュールのフラット・リストまたはフラット配列として表現することができる。この非階層システムは、コア包含フレームワーク・コードおよびそのフレームワーク内で動作するモジュールをコンパクトに保つのに役立つ。J

(27)

特表2003-502730

avaBeans (商標) コンポーネントなどの階層コンポーネントを使用するシステムは、関連する利益を提供することができるが、その利益は、より多くのシステム・リソースを必要とするより複雑な管理システムという犠牲のもとで得られる。しかし、包含フレームワークは、非階層モジュールが階層包含システムの利益の多くを得るための機構を提供する。この機構を、下で図9および10に関して説明する。

#### 【0051】

図8からわかるように、一実施形態では、中央フレームワーク・インスタンス170に、モジュールが呼び出すことができる、パブリックにアクセス可能なメソッド178が含まれる。これらのメソッドを抽象グループに分割することができる。たとえば、メソッドの1グループ172にルックアップのメソッドを含めることができる。ルックアップのメソッドでは、上で説明したルックアップ・サービスの機能性が実装される。モジュールは、中央フレームワーク・インスタンス170のルックアップ・メソッドにモジュール記述子を渡して、特定のサービス・モジュールを突き止めることができる。包含フレームワークのルックアップ・プロセスは図12に関連させて以下で説明する。フレームワーク・メソッドのもう1つのグループ174に、ロードおよびアンロードのモジュールのメソッドを含めることができる。サービス・モジュールを見つけた後に、クライアント・モジュールが、そのサービス・モジュールをロードし、ロードされたモジュールへの参照を返すように中央フレームワーク・インスタンス170に要求することができる。クライアント・モジュールは、その後、そのサービスを呼び出すことができる。クライアントは、サービス・モジュールの使用を終えた時に、そのサービス・モジュールを解放するフレームワーク・メソッドを呼び出すことができる。別個のグループとして説明したが、ルックアップ・グループおよびロード／アンロード・グループへのメソッドの分類は、概念的な分類でしかない。たとえば、一実施形態では、ルックアップ・メソッドによって、一致するモジュールをロードし、一致するモジュールへの参照を返すことができる。

#### 【0052】

図8には、フレームワーク・メタデータと称する、ロードされたモジュールの

(28)

特表2003-502730

リストを記述したデータ182およびシステムの状態を記述した他のデータを含めることができるシステム・データ180も示されている。中央フレームワーク・インスタンス170のもう1つのメソッドの抽象グループ176に、リフレクション・メソッドを含めることができる。リフレクション・メソッドは、コア・メタデータ180への直接アクセスを提供するので、他のメソッドのグループとは多少異なる。システム・モジュールと称する特殊なクラスのモジュールが、リフレクション・メソッドを呼び出して、メタデータ180へのアクセスを得ることができる。通常のモジュールは、メタデータ180にアクセスすることができない。

#### 【0053】

コア・システム・データ180への参照を得た後に、システム・モジュールは、望ましい形であればどのようにでも、データを使用するか修正することができる。したがって、包含フレームワークは非常に拡張可能である。中央フレームワーク・インスタンス170自体を小さいままにし、システム・モジュールを追加して、中央フレームワーク・インスタンス170によって既に使用可能にされてはいない機能性を実装することができる。たとえば、システム・モジュールによって、包含フレームワーク内で動作するアプリケーション／サービスと外部ネットワーク内に基づくサービスとの間で、図3から7について上で説明した統合を可能にすることができる。

#### 【0054】

この例では、そのようなシステム・モジュールを、外部ネットワークのプロトコルおよびプログラミング・モデルに準拠する副ルックアップ・サービスとして記述することができる。たとえば、J i n i (商標) ネットワークの場合、J i n i (商標) ネットワーク・ルックアップ・サービスを発見し、そのネットワーク・ルックアップ・サービスにジョインし、それ自体を副ルックアップ・サービスとして登録するシステム・モジュールを記述することができる。ネットワーク・クライアントがサービスを要求する時に、ネットワーク・ルックアップ・サービスが、そのシステム・モジュールによって実装されるルックアップ・サービスを呼び出すことができる。このシステム・モジュールは、要求されたサービスの

(29)

特表2003-502730

記述と一致する、包含フレームワーク内のサービス・モジュールを見つけることを試みることができる。一致が見つかった場合、システム・モジュールは、システム・モジュール・リストおよびメタデータへのフル・アクセスを有するので、ネットワーク・クライアントにサービス・モジュールをエクスポートするのに必要なすべてのステップを実行することができる。たとえば、システム・モジュールは、一致するサービス・モジュールをシステムにロードし、登録し、新たにロードされたモジュールへの、J a v a（商標）インターフェースなどのインターフェースを、リクエストに返すことができる。

#### 【0055】

##### 図9および10 - 階層環境のシミュレート

モジュールのための階層コンテキストを確立することが望ましいことがしばしばである。たとえば、同一タイプの複数のサービス・モジュールが、1つのシステム内に存在するが、それぞれがわずかに異なる形で振る舞う場合がある。階層包含システムでは、サービスに関するモジュールによる要求を、要求元のモジュールの親モジュールまたは包含するモジュールを介してフィルタリングし、その結果、特定のサービス・モジュールへの参照をリクエストに渡せるようにすることができる。階層包含は、モジュールの階層の間でデータを簡単に分配し、格納する能力などの、他の固有の長所も有する。しかし、上で述べたように、階層包含システムの完全な実装は、メモリおよび処理能力などの必要なシステム・リソースに関して非常にコストがかかる可能性がある。包含フレームワークは、階層包含の利益の多くを、通常はそれに関連する高いオーバーヘッド・コストなしで、開発者およびアプリケーションに与える機構を提供することができる。

#### 【0056】

たとえば、包含フレームワークの一実施形態では、モジュールが、他のモジュールのモジュール要求リスナとしてそれ自体を登録することができる。たとえば、モジュールAが、たとえば中央フレームワーク・インスタンスのA d d R e q u e s t L i s t e n e rメソッドを呼び出すことによって、それ自体をモジュールBの要求リスナとして登録することができる。モジュールBが、その後、特定のサービスを見つけるために中央フレームワーク・インスタンスのメソッドを

(30)

特表2003-502730

呼び出す時に、中央フレームワーク・インスタンスが、モジュールBのモジュール要求リスナのすべてを検査する。この場合、中央フレームワーク・インスタンスは、要求リスナとしてモジュールAを見つけ、モジュールAに、要求されたサービス・モジュールをモジュールBに提供するように要求する。

#### 【0057】

図9および10に、包含フレームワークでのモジュール要求リスナの例示的使用を示す。図9に、印刷サービスに関する所望の概念的なモジュール階層を示す。図からわかるように、2つの印刷サービス・モジュール192および194、すなわち印刷サービスAおよび印刷サービスBが、印刷マネージャ・モジュール190内でカプセル化されている。たとえば、2つの印刷サービス192および194が、異なる位置に印刷し、異なる解像度およびカラー機能などを有することができる。これらの印刷サービス・モジュールのいずれかが、印刷サービスに関する別のモジュールによって行われるルックアップ要求を満足することができる。しかし、特定の印刷サービスを選択し、返す、印刷マネージャ・モジュールを使用することが望ましい場合がある。たとえば、印刷マネージャ190が、どのクライアント・モジュールが印刷要求を行うかに基づいて印刷サーバを選択することができる、また、印刷マネージャが、所望の印刷サービス特性に関するユーザ入力を求めるダイアログ・ボックスを表示することができる。

#### 【0058】

包含フレームワークでは、非階層包含モデルが使用されるが、印刷マネージャ・モジュール190を、印刷サービスを要求する可能性があるクライアント・モジュールのモジュール要求リスナとして登録することによって、図9に示された階層を実現することができる。図10に、システム内で動作することができる例のモジュール198を示す。前に説明したように、これらのモジュールは、それ自体が、他のモジュールをサービスとして使用することができる。包含フレームワークの非階層モデルに従って、モジュールが、固有のモジュール階層なしで、フラットなレイアウトに配置されて図示されている。

#### 【0059】

この例では、ウェブ・ブラウザ・モジュール196を、たとえばウェブ・ペー

(31)

特表2003-502730

ジを印刷するために、印刷要求を行うように動作可能とすることができる。図10からわかるように、印刷マネージャ・モジュール190を、ウェブ・ブラウザ・モジュール196のモジュール要求リスタとして登録することができる。ウェブ・ブラウザ196から印刷サービス要求を受け取った時に、包含フレームワーク・ルックアップ・サービスが、ウェブ・ブラウザ・モジュール196の要求リスタとして登録された印刷マネージャ・モジュール190を見つけることができ、ウェブ・ブラウザ・リクエスト196に印刷サービス・モジュールを提供するように印刷マネージャ・モジュール190に要求することができる。その後、印刷マネージャ・モジュール190が、印刷サービス・モジュールA192または印刷サービス・モジュールB194への参照を返すことができ、もしくは、印刷マネージャ・モジュール190が、どの印刷サービス・モジュールを返すかを決定するためにユーザにダイアログ・ボックスを提示するなどを行うことができる。したがって、図9の所望のモジュール階層を、包含フレームワークの非階層モジュールのために実施することができる。

#### 【0060】

##### 図11 - パーセル・パッケージ化ユニット

モジュールを、パーセルと称するユニットにパッケージ化することができる。このパッケージ化は複数の目的のために働く。たとえば、パーセルは、関連するコードおよびデータをユニットとして管理する便利な機構を提供する。密に関連するモジュールが静的依存性を有する場合に、それらを一緒にパーセルにパッケージ化することができる。パーセルは、システム内のインストールおよびアップグレードを扱うのに使用することができる。

#### 【0061】

図11に、個人情報マネージャ（PIM）に関連するメソッドを一緒にグループ化する例のパーセル200を示す。この図には、カレンダー・モジュール202、連絡先リスト・モジュール204、面会モジュール208、およびユーザ・インターフェース・モジュール206が示されている。さまざまな他のモジュールが、望みに応じてパーセル内に存在することができる。PIMパーセル200のモジュールは、ブックマーク・サービス、検索サービスなど、包含フレームワー

(32)

特表2003-502730

ク内で動作するさまざまなコア・サービス・モジュールを利用することもできる。PIMパーセルを使用することによって、PIMアプリケーションのインストールおよびアップグレードを単純にすることができる。PIMモジュールのこの形でのパーセルへのパッケージ化は、複数ターゲット開発用の別々のコード・ユニットの作成という開発時の利益も有する。

#### 【0062】

パーセルは、非階層モジュールのランタイム・コンテキストを提供する他の方法も提供する。モジュールがシステムにロードされる時に、中央フレームワーク・インスタンスが、パーセルがある場合に、モジュールがどのパーセルに属するかを指定するメタデータを記憶することができる。サービス・モジュールは、後に、この情報を使用して、クライアントが属するパーセルに応じて、異なるクライアント・モジュールに異なる形でサービスを提供することができる。たとえば、クライアント・モジュールが、ファイル・アクセス・サービス・モジュールを使用して、ルート・ディレクトリを得ることができる。ファイル・アクセス・モジュールは、そのクライアントが属するパーセルに応じて、異なるクライアントに異なるルート・ディレクトリを返すことができる。

#### 【0063】

##### 図12 - モジュール要求の流れ図

図12は、中央フレームワーク・インスタンスが、クライアント・モジュールからサービス・モジュールに関するルックアップ要求を受け取った時に実行することのできる、通常のルックアップ・プロセスを示す流れ図である。図12は、例示的であり、さまざまなステップを組み合わせるか、省略するか、修正することができることに留意されたい。たとえば、前に注記したように、ルックアップ・プロセスをカスタマイズするシステム・モジュールを追加することができる。

#### 【0064】

図12のステップ300で、中央フレームワーク・インスタンスが、リクエスト・モジュールからモジュール・ルックアップ要求を受け取る。たとえば、リクエスト・モジュールは、中央フレームワーク・インスタンスのRequestModuleメソッドを呼び出し、要求されるサービス・モジュールのモジュール



(33)

特表2003-502730

記述子ならびにリクエスト・モジュール自体への参照を渡すことができる。リクエスト・モジュールへの参照をシステム・データに追加し、したがって、サービス・モジュール・ユーザを記憶することができる。下で詳細に説明するように、モジュールを、他のモジュールがそれを使用していない時にアンロードすることができる。

#### 【0065】

リクエスト・モジュールが渡すモジュール記述子によって、フレームワーク・インスタンスが一致するモジュールの検索を試みるのに使用することができる、要求されるモジュールに関するさまざまな属性が指定される。このモジュール記述子は、要求されるモジュールのサービス・タイプ、クラス名、またはサービス固有の属性などの情報を含むオブジェクトとすることができる。リクエストは、中央フレームワーク・インスタンスにテキスト記述を渡すこともでき、中央フレームワーク・インスタンスは、これを使用してモジュール記述子オブジェクトを作成することができる。

#### 【0066】

ステップ302で、中央フレームワーク・インスタンスが、要求元モジュールに関して要求リスナ・モジュールが登録されているかどうかを調べるために検査する。要求リスナが見つかる場合には、ステップ304で、フレームワーク・インスタンスが、要求について要求リスナに通知し、モジュール要求記述子に一致するモジュールの供給を試みるように要求リスナに指示する。要求リスナが、一致するモジュールを供給できる場合には、実行はステップ314に進む。そうでない場合には、一致が見つかるか要求リスナがなくなるまで、他の登録された要求リスナに、モジュールの供給を求めることができる。

#### 【0067】

要求リスナが見つからない場合、またはどの要求リスナも要求されたモジュールを供給できない場合に、実行はステップ306に進む。しかし、一実施形態では、1つまたは複数の要求リスナが要求元モジュールについて登録されており、そのどれかが一致するモジュールを供給できない場合に、ステップ304の後に実行を停止することができる。ステップ306では、中央フレームワーク・イン

(34)

特表2003-502730

スタンスが、モジュールのリストを検査して、モジュールの1つがモジュール記述子に一致するかどうかを判定する。一致が見つかる場合には、ステップ308で、フレームワーク・インスタンスが、一致したモジュールが複数インスタンス化可能であるかどうかを判定する。そうでない場合には、実行はステップ314に進む。

**【0068】**

ステップ308で、一致したモジュールが複数インスタンス化可能であることがわかった場合には、中央フレームワーク・インスタンスは、一致に関するモジュール・リストの検索を継続することができる。検索すべきモジュールがもうない場合には、実行はステップ310に進む。ステップ310で、フレームワーク・インスタンスが、モジュール・リスト内でモジュールプロバイダ・モジュールを検索する。モジュールプロバイダ・モジュールは、要求されたモジュールを供給することができるモジュールである。たとえば、ネットワーク・ルックアップ・サービスを、包含フレームワークのモジュールプロバイダ・モジュールとしてインポートすることができる。

**【0069】**

モジュールプロバイダ・モジュールが見つかった場合には、ステップ312で、中央フレームワーク・インスタンスが、要求についてモジュールプロバイダ・モジュールに通知し、モジュール要求記述子に一致するモジュールの供給を試みるように指示する。一致が見つかる場合には、実行はステップ314に進む。モジュール・プロバイダが、要求されたモジュールを供給できない場合には、中央フレームワーク・インスタンスは、他のモジュールプロバイダ・モジュールを検索し、ステップ312を繰り返すことができる。モジュール・リストにモジュール・プロバイダが存在しない場合、またはどれもが要求されたモジュールを供給できない場合には、リクエストが、要求を満足できないことを通知され、実行が完了する。

**【0070】**

ステップ314は、ステップ304、308、または312から到達することができる。どの場合でも、モジュール要求記述子に一致するモジュールが見つかる。

(35)

特表2003-502730

る。ステップ314で、リクエストが、一致するモジュールのユーザとして登録され、ステップ316で、一致するモジュールへの参照が、リクエストに返される。一致するモジュールのロードおよび初期化に伴う必要な初期化ステップも、ステップ314で実行される。たとえば、モジュールが、モジュールがロードされた時に呼び出されるInitializeメソッドを有する場合がある。

#### 【0071】

上で注記したように、図12の流れ図は、例示的であり、さまざまな実施形態が、異なるルックアップ／ロードのシナリオを有する可能性がある。たとえば、モジュールが、一致するモジュールへの参照を返さずにサービス・モジュールをロードするために中央フレームワーク・メソッドを呼び出すことができ、また、場合によっては、要求リスナを無視することができる。

#### 【0072】

##### 図13 - モジュール解放の流れ図

クライアント・モジュールが、サービス・モジュールの使用を終了した時に、クライアントは、中央フレームワーク・インスタンスのメソッドを呼び出して、モジュールを解放することができる。図13は、モジュール解放プロセスを示す流れ図である。図13の流れ図は、例示的であり、異なる実施形態に関する必要または望みに応じて、さまざまなステップを組み合わせるか、省略するか、追加するか、修正することができる。

#### 【0073】

ステップ330で、中央フレームワーク・インスタンスが、ユーザ・モジュールからモジュール解放通知を受け取る。上で図12に関して説明したように、ユーザ・モジュールがサービス・モジュールを要求する時に、ユーザ・モジュールが、サービス・モジュールのユーザのリストに追加される。ステップ332で、中央フレームワーク・インスタンスが、解放を行っているユーザ・モジュールを、解放されるモジュールのユーザのリストから除去する。ステップ334で、フレームワーク・インスタンスが、他のユーザ・モジュールが解放されるモジュールを使用しているかどうかを、たとえば他のモジュールが解放されるモジュールのユーザ・モジュール・リストに存在するかどうかを検査することによって、判

(36)

特表2003-502730

定する。そうである場合には、実行が停止する。

#### 【0074】

他のモジュールが、解放されるモジュールを使用していない場合には、中央フレームワーク・インスタンスは、解放されるモジュールのアンロードを試みることができる。ステップ336で、フレームワーク・インスタンスが、解放されるモジュールのCanFinalizeメソッドを呼び出すことができる。CanFinalizeメソッドは、モジュールをアンロードできる場合に真、そうでない場合に偽を返す。ステップ338で、CanFinalizeメソッドが偽を返す場合には、実行が停止する。そうでない場合には、解放されるモジュールのFinalizeメソッドを呼び出すことができる。Finalizeメソッドは、リソースの解放など、モジュールのアンロードに関する必要なステップを実行することができる。その後、モジュールをアンロードすることができ、これには、特定の実施形態に応じて、ガーベジ・コレクションなどが含まれる可能性がある。

#### 【0075】

特定の実施形態に関して本発明を説明してきたが、本明細言に記載の特定の形態に制限することは意図されておらず、逆に請求項によって定義される本発明の趣旨および範囲に正当に含めることができる代替形態、修正形態、および同等物を含むことが意図されている。

#### 【図面の簡単な説明】

##### 【図1】

通常の小フットプリント・デバイスのハードウェア・アーキテクチャを示すブロック図である。

##### 【図2】

包含フレームワーク内でアプリケーションとサービスを実行するシステムで用いられるハードウェア／ソフトウェア層の通常の階層を示す図である。

##### 【図3】

包含フレームワーク内でアプリケーション／サービスを実行する小フットプリント・デバイスが、ローカル・サービスベース・ネットワークに接続される、例

(37)

特表2003-502730

示的ネットワークを示す図である。

【図4】

サービス・プロバイダがルックアップ・サービスを見つける、ディスカバリ・プロセスを示す図である。

【図5】

サービス・プロバイダがそのサービスをルックアップ・サービスに登録する、ジョイン・プロセスを示す図である。

【図6】

クライアントがルックアップ・サービスにサービスを要求する、ルックアップ・プロセスを示す図である。

【図7】

クライアントが、ルックアップ・サービスから受け取ったサービス・オブジェクトを使用してサービスを呼び出す、サービス呼出しプロセスを示す図である。

【図8】

包含フレームワークの基本アーキテクチャを示す抽象ブロック図である。

【図9】

階層包含環境をシミュレートするための、包含フレームワークでのモジュール要求リスナの使用を示す図である。

【図10】

階層包含環境をシミュレートするための、包含フレームワークでのモジュール要求リスナの使用を示す図である。

【図11】

モジュールを一緒にグループ化するためのパーセルの使用を示す図である。

【図12】

中央フレームワーク・インスタンスが、クライアント・モジュールからサービス・モジュールに関するルックアップ要求を受け取った時に実行することのできる、通常のルックアップ・プロセスを示す流れ図である。

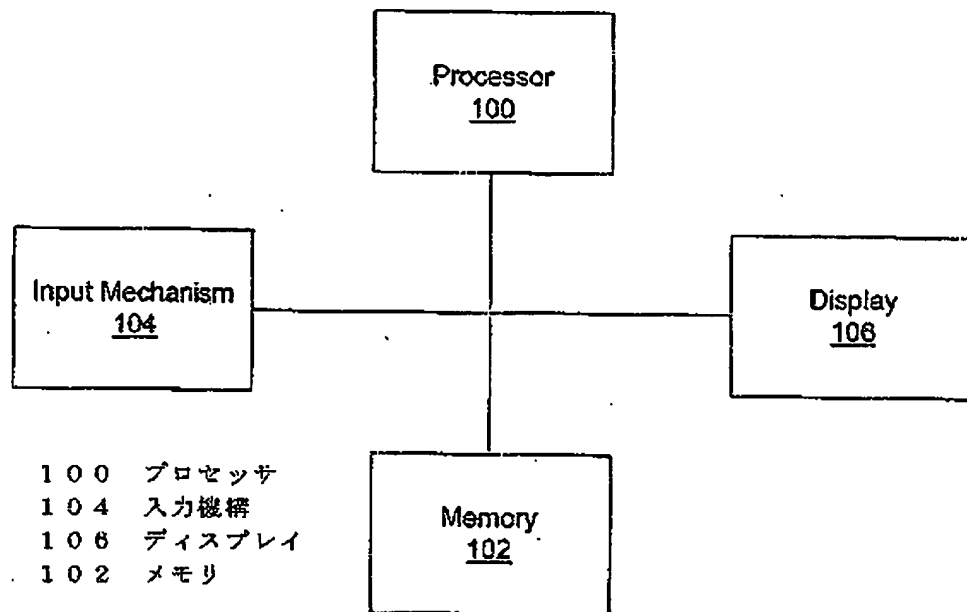
【図13】

モジュール解放プロセスを示す流れ図である。

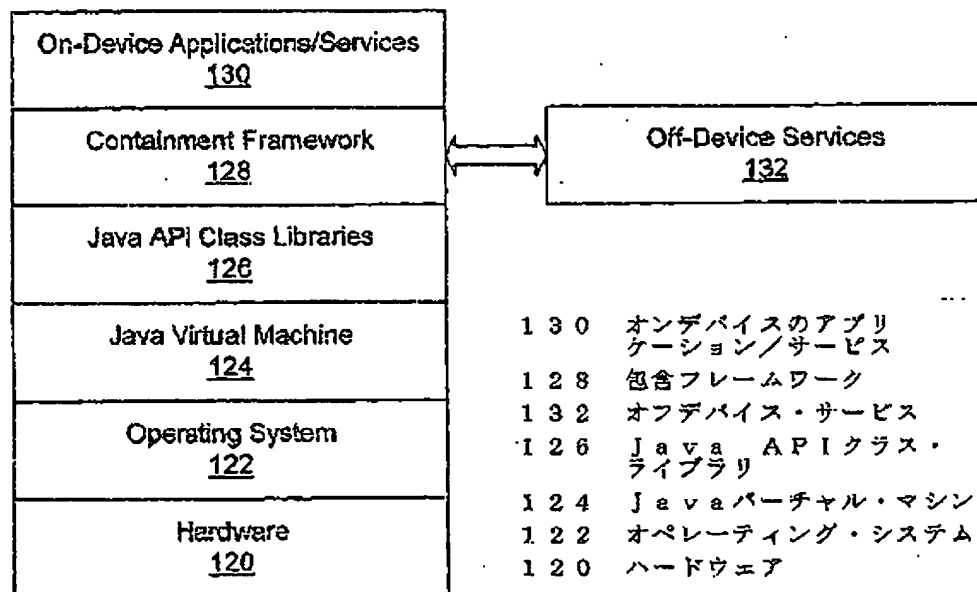
(38)

特表2003-502730

【図1】



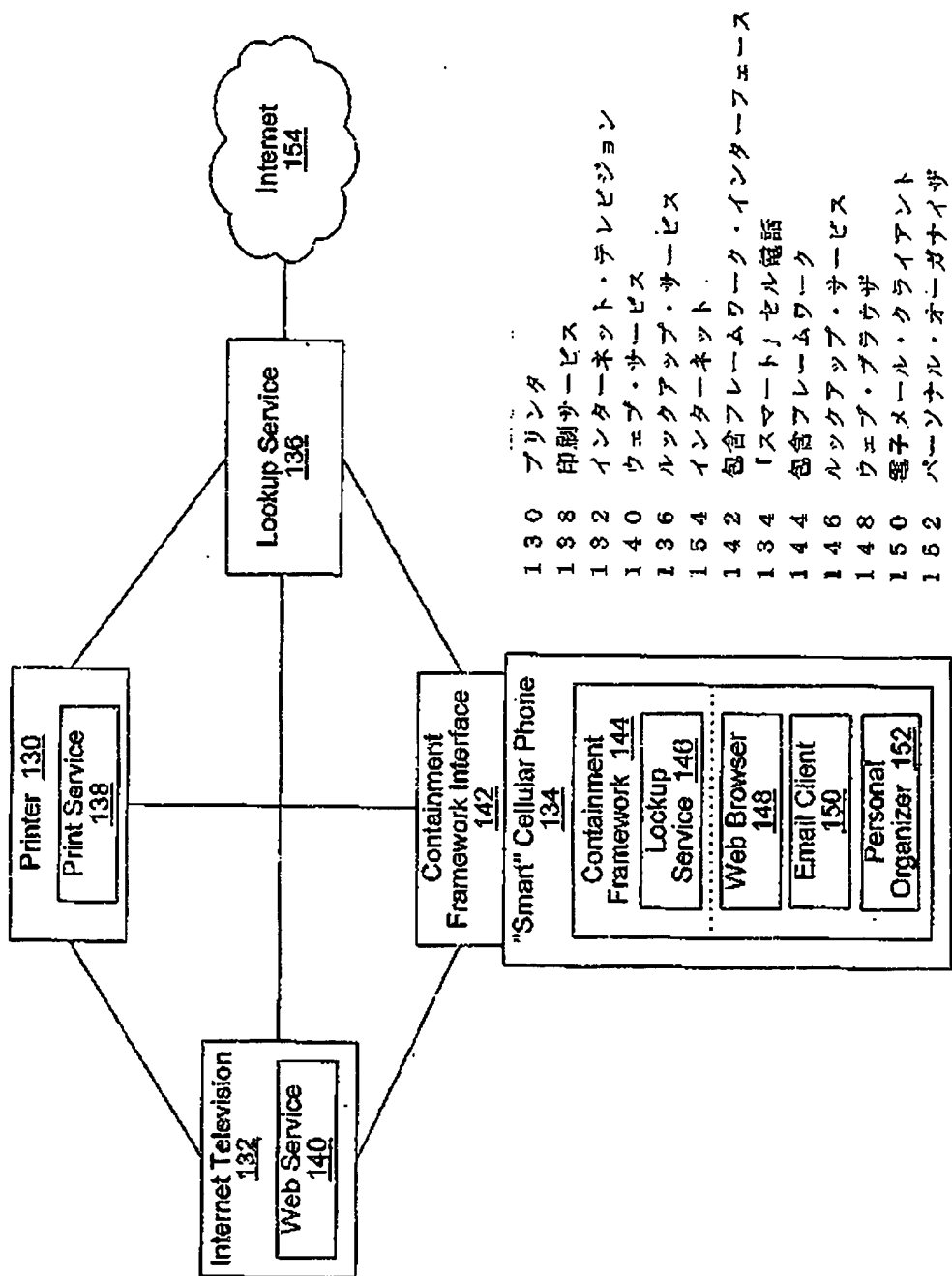
【図2】



(39)

特表2003-502730

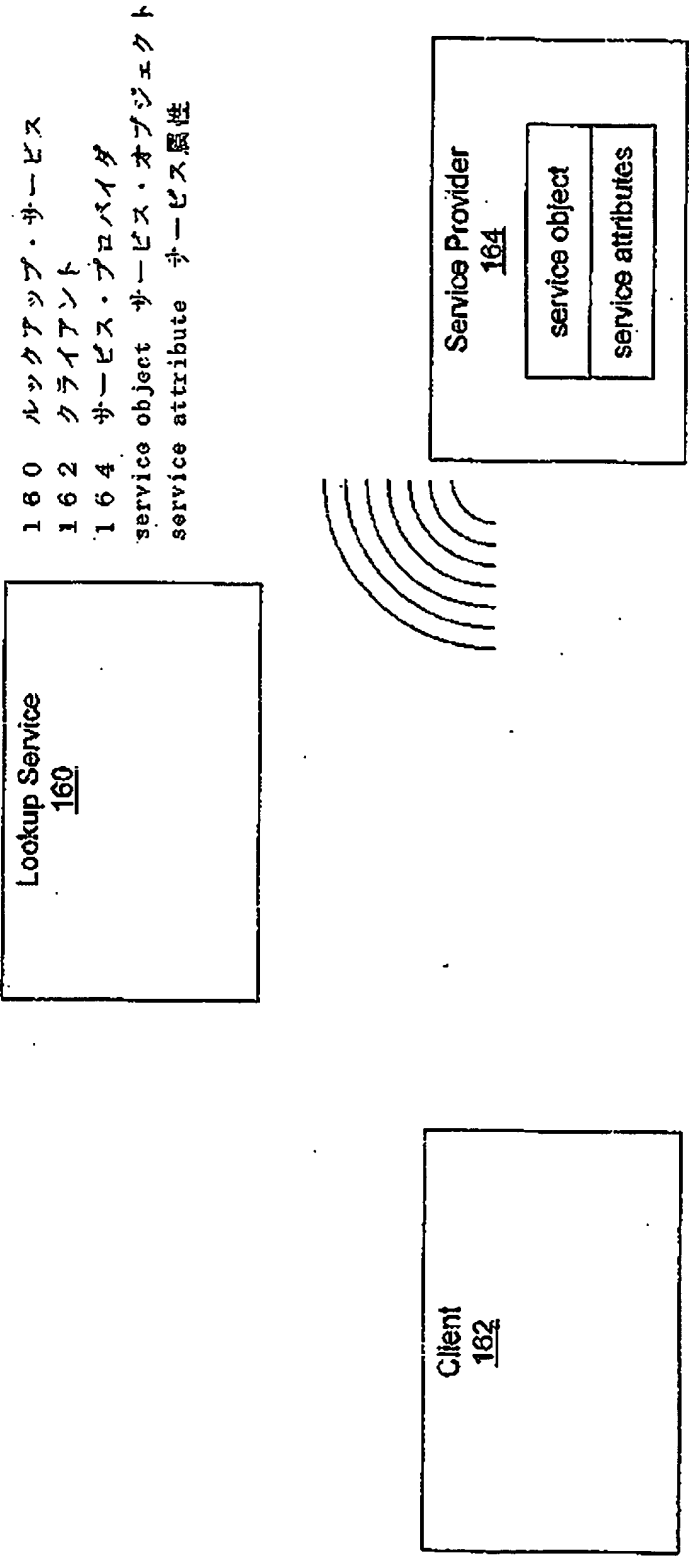
【図 3】



(40)

特表2003-502730

【図 4】

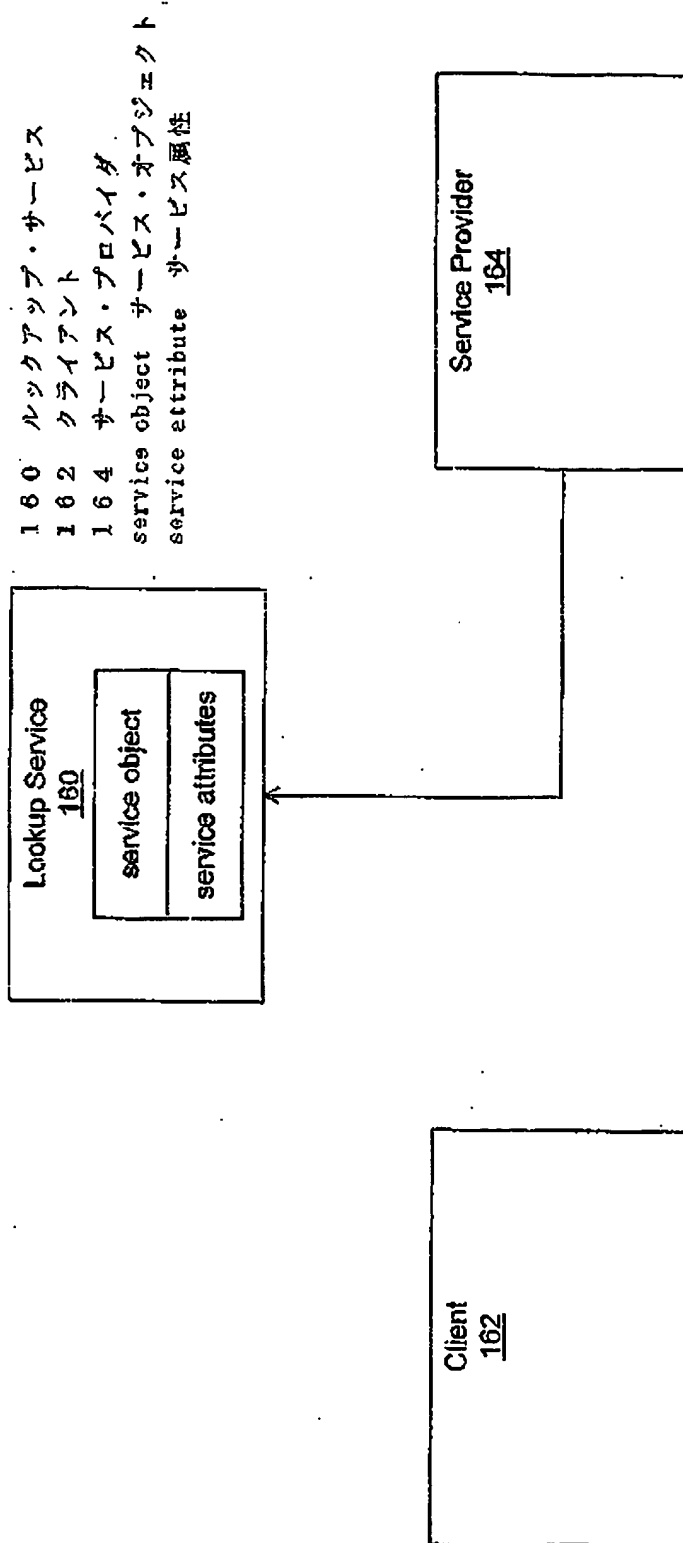




特表2003-502730

(41)

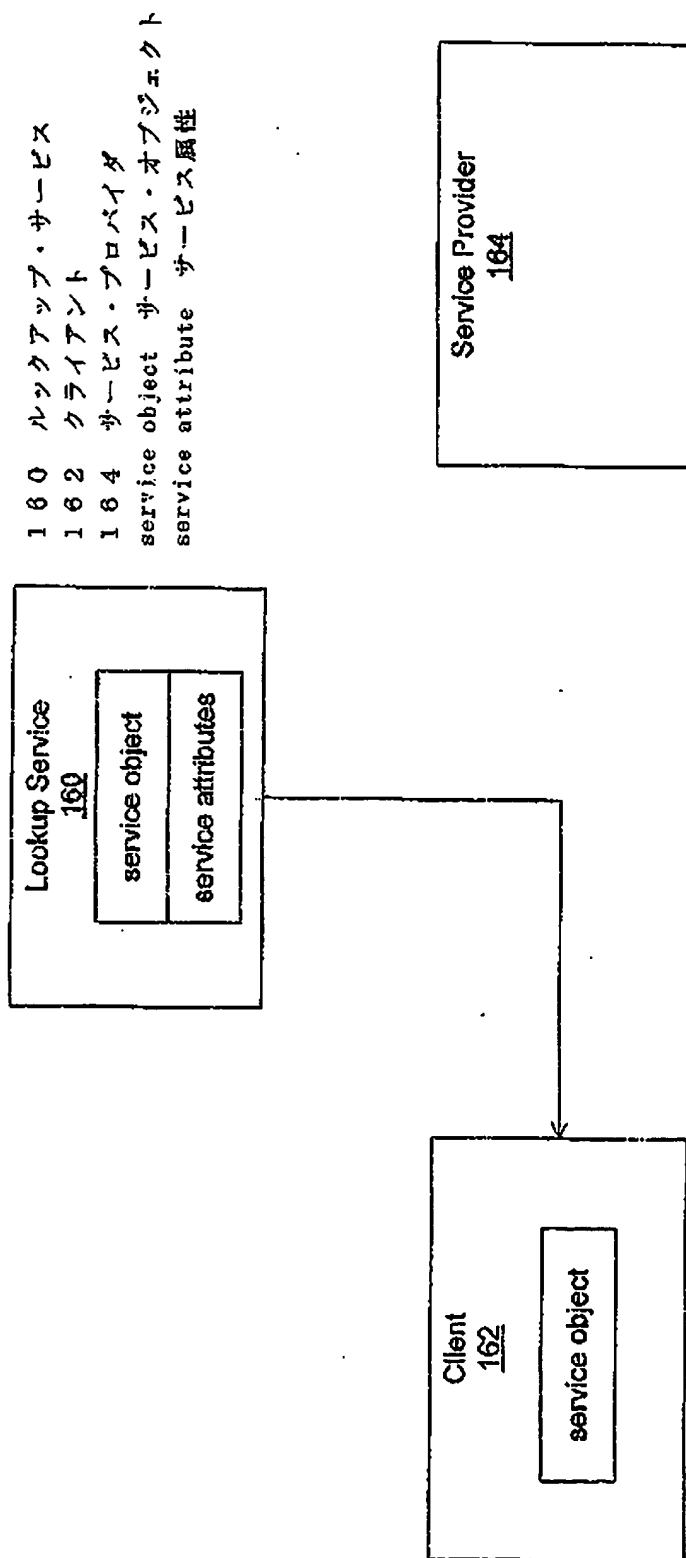
【図5】



特表2003-502730

(42)

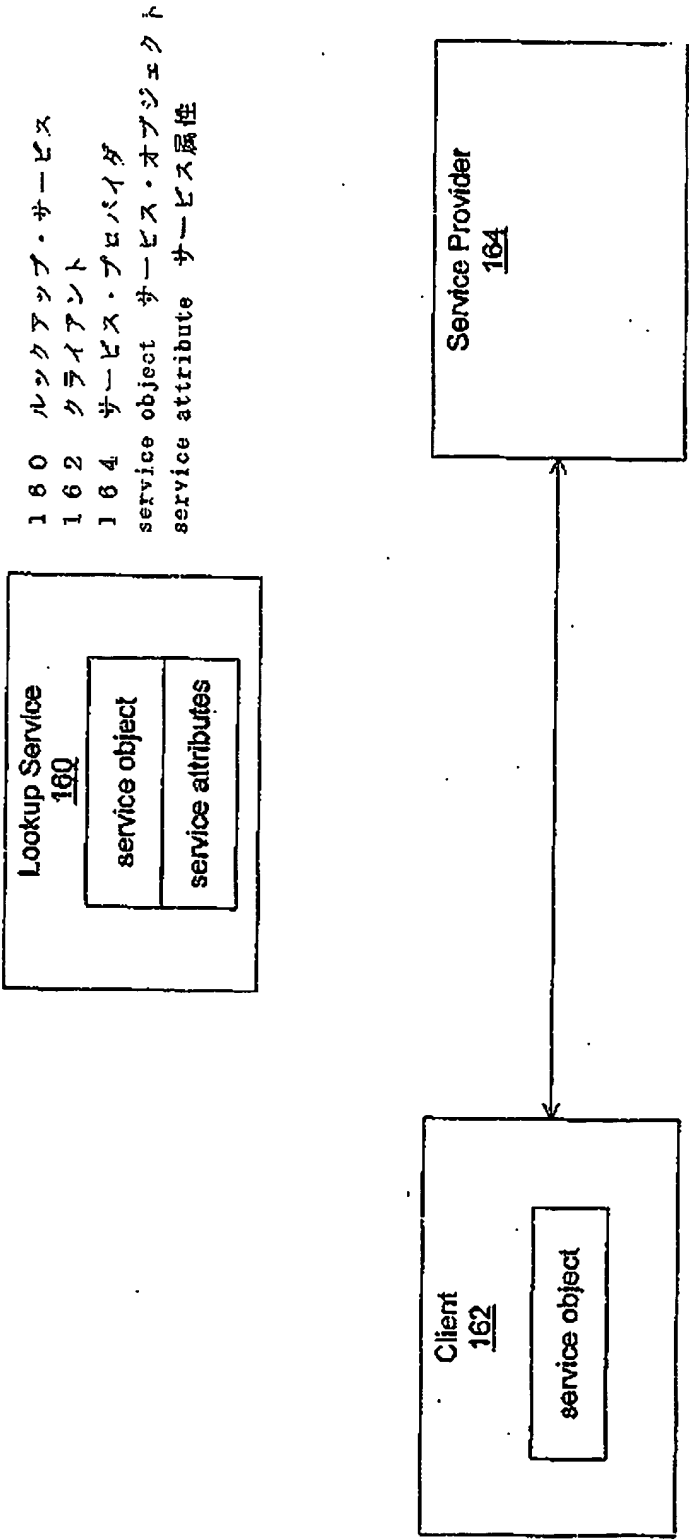
【図6】



(43)

特表2003-502730

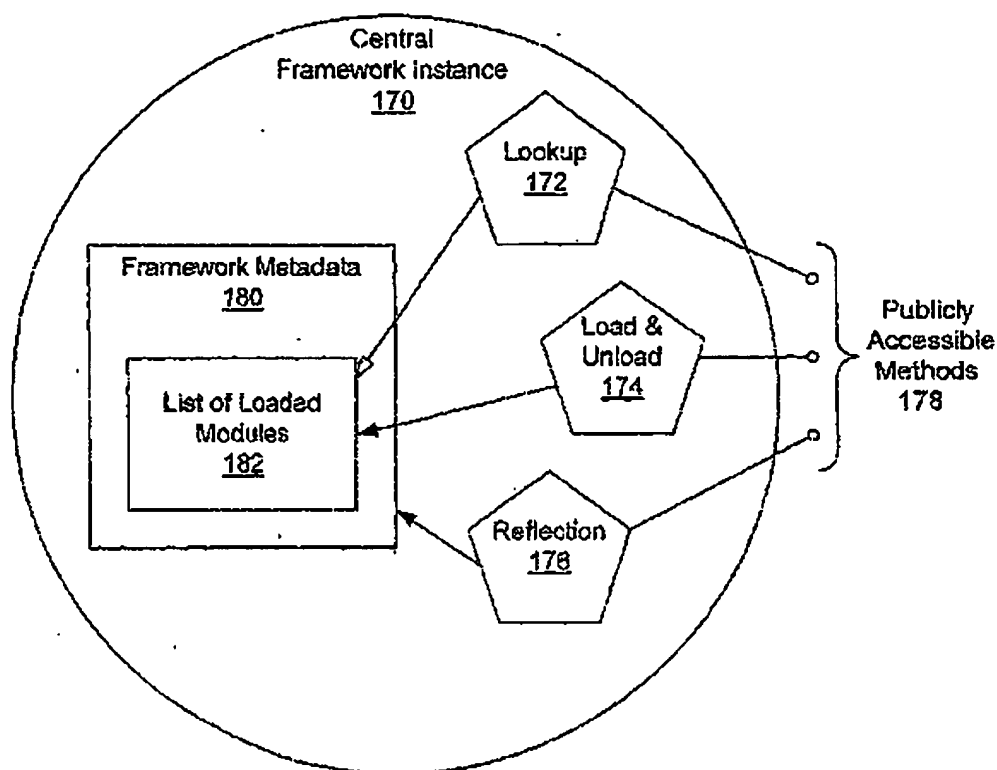
【図7】



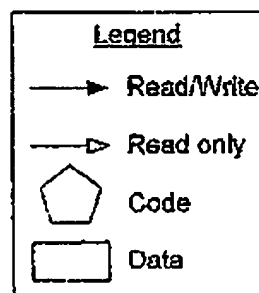
(44)

特表2003-502730

【図 8】



170 中央フレームワーク・インスタンス  
 180 フレームワーク・メタデータ  
 182 フロッドアップされたモジュールのリスト  
 172 ロードおよびアンロード  
 174 リフレクション  
 176 パブリックにアクセス可能なメソッド  
 Legend  
 Read/Write 読み書き  
 Read only 読み専用  
 Code 例  
 Data コード

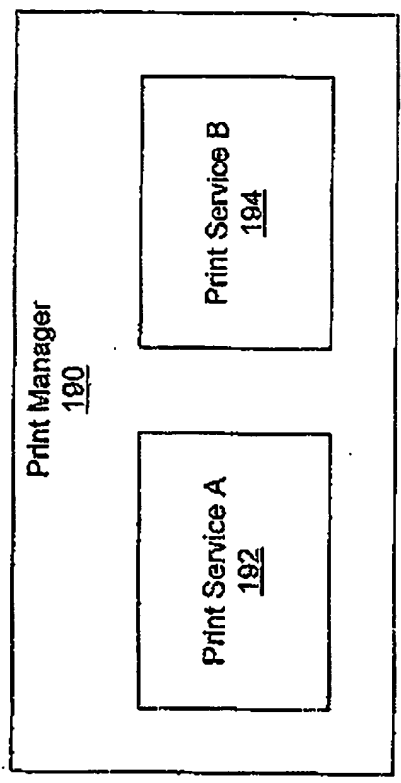


(45)

特表2003-502730

【図9】

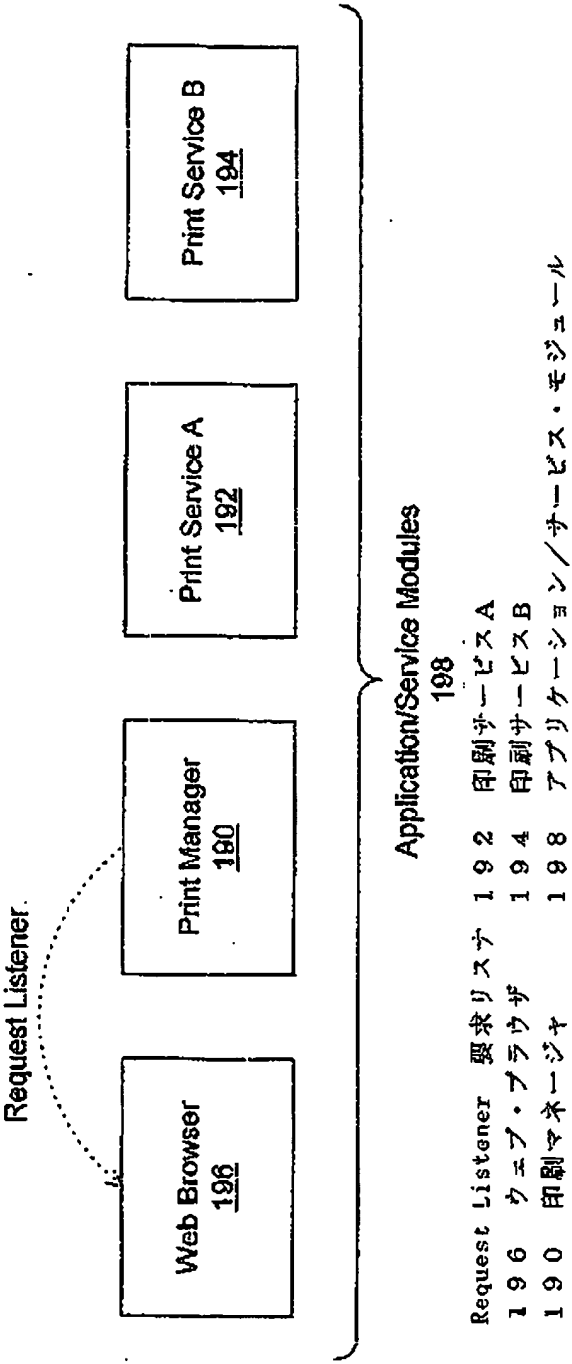
190 印刷マネージャ  
192 印刷サービスA  
194 印刷サービスB



特表2003-502730

(46)

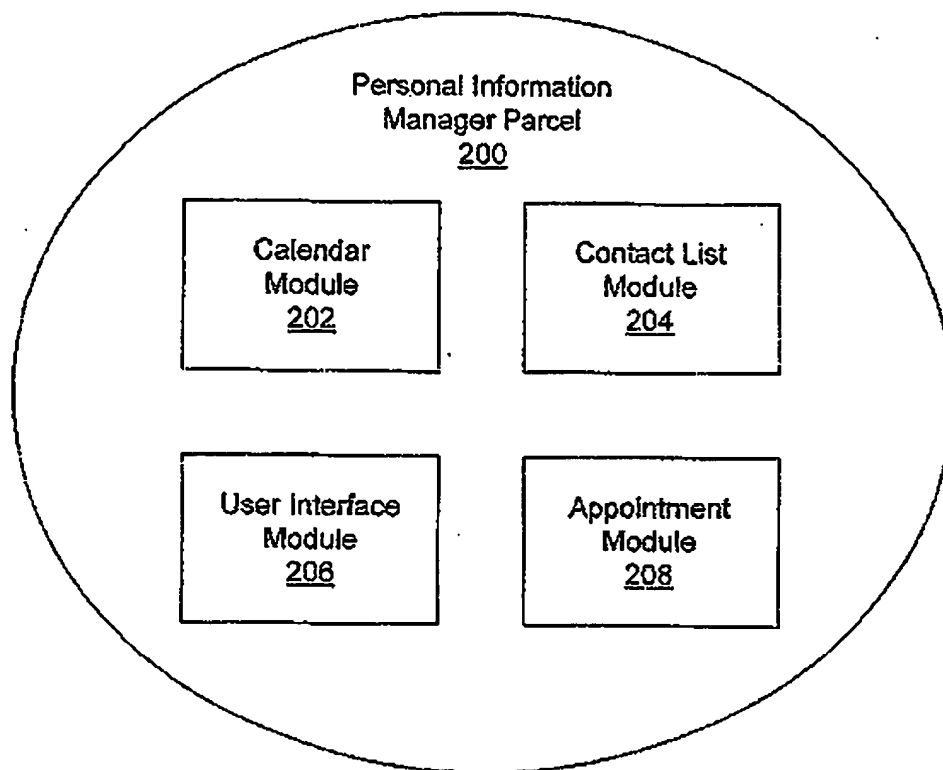
【図10】



(47)

特表2003-502730

【図11】

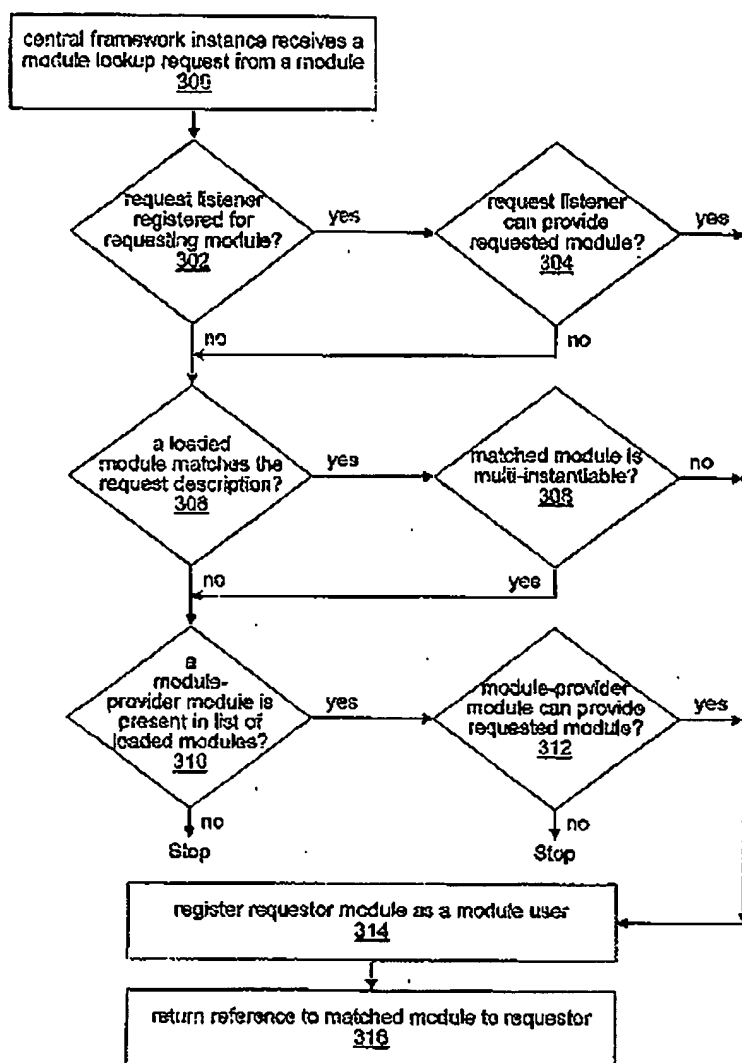


- 200 個人情報マネージャ・パーセル
- 202 カレンダ・モジュール
- 204 連絡先リスト・モジュール
- 206 ユーザ・インターフェース・モジュール
- 208 面会モジュール

(48)

特表2003-502730

【図12】



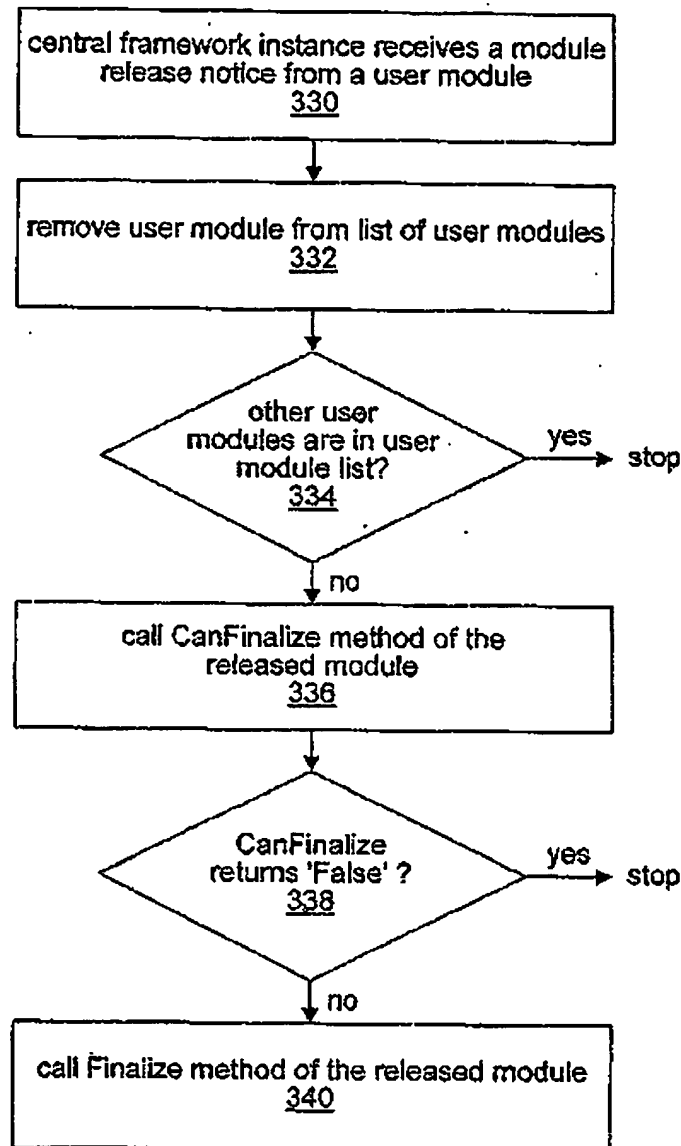
300 中央フレームワーク・インスタンスがモジュールからモジュール・  
 302 レッジャーに登録されているか  
 304 要求されたモジュールが登録されているか  
 306 一致したモジュールが複数インスタンス化可能であるか  
 308 一致したモジュールがロードされたモジュールのリストにあるか  
 310 モジュールプロバイダ・モジュールが要求されたモジュールを供給できるか  
 312 モジュールが要求されたモジュールを供給できるか  
 314 一致したモジュールへの参照をリクエストに返す



(49)

特表2003-502730

【図13】



330 中央フレームワーク・インスタンスが、ユーザ・モジュールから  
 332 モジュール解放通知を受け取る  
 334 ユーザ・モジュールをユーザ・モジュールのリストから除去する  
 336 他のユーザ・モジュールが、ユーザ・モジュール・リストにあるか  
 338 解放されるモジュールのCanFinalizeメソッドを呼び出す  
 340 CanFinalizeメソッドが「偽」を返すか  
 340 解放されるモジュールのFinalizeメソッドを呼び出す

## INTERNATIONAL SEARCH REPORT

# INTERNATIONAL SEARCH REPORT

Inventor: T. Application No.

PCT/US 00/14173

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 606F9/46 606F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS OF SEARCHED

Primary document classification (classification system followed by classification symbols)

IPC 7 606F

Documentation searched for: other than main documentation to the extent that such documents are indicated in the fields searched

Electronic data base consulted during the international search phase or data base and, where possible, search terms used

INSPEC, EPO-Internal, IBM-ROB, MPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Character of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DOLVANE J ET AL: "PersonalJava & information appliances. 1" DR. DOBB'S JOURNAL, JAN. 1999, MILLER FREEMAN, USA, vol. 24, no. 1, January 1999 (1999-01), pages 60-61, 64 - 68, XP000951660 ISSN: 1044-789X the whole document	1-30
A	CLARK D: "Network nirvana and the intelligent device" IEEE CONCURRENCY, APRIL-JUNE 1999, IEEE, USA, vol. 7, no. 2, April 1999 (1999-04) - June 1999 (1999-06), pages 16-19, XP002327392 ISSN: 1092-3063 the whole document.	1-30

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"B" earlier document but published on or after the international filing date

"C" document which may show doubts on priority claim of which the date is established by the publication date of another citation or other appropriate reference (as reported)

"D" document referring to an oral disclosure, use, exhibition or other means

"E" documents published prior to the international filing date but later than the priority date claimed

"F" later document published after the international filing date or priority date and not in conflict with the application but also fundamental to the principle or theory underlying the invention

"G" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"H" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is considered with one or more other such documents, such combination being obvious to a person skilled in the art.

"I" document member of the same patent family

Date of the actual completion of the international search

10 October 2000

Date of issuing of the international search report

06/11/2000

Name and mailing address of the ISA  
European Patent Office, P.O. Box 2911, 8001  
NL - 2200 HV Rijswijk  
Tel.: +31-70-340-3400, Telex: 510 401  
Fax: +31-70-340-3010

Authorized officer

Fonderson, A

(51)

特表2003-502730

## INTERNATIONAL SEARCH REPORT

 Inventor at Application No.  
 PCT/US 00/14173

C (Classification) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	STEVEN FRANK: "PersonalJava and Inferno" JAVA DEVELOPER'S JOURNAL, SYS-CON PUBLICATIONS, USA, vol. 3, no. 3, March 1998 (1998-03), pages 8-14, XF002149692 ISSN: 1087-6944 the whole document	1-30
A	FUENTES L ET AL: "A Java framework for Web-based multimedia and collaborative applications" IEEE INTERNET COMPUTING, MARCH-APRIL 1999, IEEE, USA, vol. 3, no. 2, pages 55-64, XP002149695 ISSN: 1089-7801 page 56, left-hand column, paragraph 2 -page 61, left-hand column, paragraph 2; figures 1,5,6	1-30
A	WD 98 37486 A (EIRICH THOMAS ;IBM (US)) 27 August 1998 (1998-08-27)  page 3, line 12 -page 9, last line	1-10, 15-24, 27-30

2

Form PCT-ISA/B (2) (Part II) of September 1998

page 2 of 2

(52)

特表2003-502730

## INTERNATIONAL SEARCH REPORT

... (information on patent family members)

International Application No.

PCT/US 00/14173

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
NO 9637486 A	27-03-1998	NONE	

Form PCT/ISA/210 (Information on patent family members)

(53)

特表2003-502730

---

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW

(72)発明者 タガン, ジョッシュ

カナダ国・ケイ1エヌ 5ジー3・オンタ  
リオ・オタワ・セント アンドリュー ス  
トリート・153-205

Fターム(参考) 5B076 BA04 BB02 BB04 BB06 DD06

【要約の続き】

ームワーク上に構築されたオンデバイス・アプリケーション/サービスを、ネットワーク・サービス・プロバイダで動作するオフデバイス・サービスと統合することができる。